RTP Payload Format for Visual Volumetric Video-based Coding (V3C)

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on 29 December 2022.

Copyright Notice

Table of Contents

1. Introduction

Volumetric video, similar to traditional 2D video, when
uncompressed, is represented by a large amount of data. 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 encoder converts volumetric frames, 3D volumetric information,
into a collection of 2D images and associated data, known as atlas
data. The converted 2D images are subsequently coded using existing
video or image codecs, e.g. ISO/IEC International Standard 14496-10
[ISO.IEC.14496-10], ISO/IEC International Standard 23008-2 [ISO.IEC.
23008-2] or ISO/IEC International Standard 23090-3 [ISO.IEC.
23090-3]. The atlas data is coded with mechanisms specified in
[ISO.IEC.23090-5]. V3C is generic mechanism for volumetric video
coding and it can be used by applications targeting volumetric
content, such as point clouds, immersive video with depth, mesh
representations of visual volumetric frames, etc. Examples of such
applications are Video-based Point Cloud Compression (V-PCC)
[ISO.IEC.23090-5], and MPEG Immersive Video (MIV) [ISO.IEC.
23090-12].

V3C utilizes high level syntax (HLS) syntax design known from
traditional 2D video codecs to represent the associated coded data,
i.e. atlas data. The atlas data is represented by Network
Abstraction Layer (NAL) units. Consequently, RTP payload format for
V3C atlas data described in this memo shares design philosophy,
security, congestion control, and overall implementation complexity
with other the 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", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in [RFC2119].

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

3. Definitions, and Abbreviations

3.1. Definitions

3.1.1. General

This document uses the definitions of [ISO.IEC.23090-5]. The following terms, defined in [ISO.IEC.23090-5], are provided here for convenience:

3.1.2. 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 CASs.

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.

atlas frame parameter set: syntax structure containing syntax elements that apply to zero or more entire coded atlas frames as determined by the content of a syntax element found in each tile header.

atlas sequence parameter set: syntax structure containing syntax elements that apply to zero or more entire coded atlas sequences as determined by the content of a syntax element found in the AFPS referred to by a syntax element found in each tile header.

attribute: scalar or vector property optionally associated with each point in a volumetric frame such as colour, reflectance, surface normal, time stamps, material ID, etc.

coded atlas sequence: sequence of coded atlas access units, 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.

intra random access point coded atlas: coded atlas for which each ACL NAL unit has nal_unit_type in the range of NAL_BLA_W_LP to NAL_RSV_IRAP_ACL_29, inclusive.

intra random access point coded atlas access unit: access unit in which the coded atlas with nal_layer_id equal to 0 is a IRAP coded atlas.

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 is either empty or has the form of a string of data bits containing syntax elements followed by an RBSP stop bit and zero or more subsequent bits equal to 0.

tile: independently decodable rectangular region of an atlas frame

visual volumetric video-based coding atlas sub-bitstream: extracted sub-bitstream from the V3C bitstream containing whole or portion of an atlas bitstream.

visual volumetric video-based coding video sub-bitstream: extracted sub-bitstream from the V3C bitstream containing whole or portion of an video bitstream.

visual volumetric video-based coding component: atlas, occupancy, geometry, or attribute of a particular type that is associated with a V3C volumetric content representation.

visual volumetric video-based coding 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.

3.1.3. Definitions Specific to This Memo

   Placeholder

3.2. Abbreviations

   ACL atlas coding layer

   AFPS atlas frame parameter set

   AP aggregation packet

   ASPS atlas sequence parameter set

   AU aggregation unit

   CAS coded atlas sequence

   DON decoding order number

   IRAP intra random access point

   MRMT Multiple RTP streams on Multiple media Transports

   MRST Multiple RTP streams over a Single media Transport

   MTU maximum transmission unit

   NAL network abstraction layer

   NALU NAL unit

   RBSP raw byte sequence payload

   SRST Single RTP stream on a Single media Transport

   V3C visual volumetric video-based coding

   VPS V3C parameter set

4. Media Format Description

4.1. Overview of the V3C codec

   ISO/IEC International Standards 23090-5 [ISO.IEC.23090-5] enables
   encoding and decoding processes of volumetric video which utilizes
   2D video coding technologies and associated data. V3C encoding of
   volumetric frame is achieved through a conversion of volumetric
   frame from its 3D representation to multiple 2D representations and
   a generation of associated data.

2D representations, known as V3C video components, of volumetric frame are encoded using traditional 2D video codecs. 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 properties of that voxel, e.g. color or material information.

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 volumetric frame. Atlas data is composed of a collection of patches. Each patch identifies a region in all V3C video components and provides information necessary to perform the appropriate inverse projection of the indicated region back into 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 decodeable. 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. V3C 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. V3C unit payload contains V3C video components, V3C atlas components or V3C parameter set. V3C video component, i.e. occupancy, geometry, and attribute, corresponds to video data units (e.g. NAL units defined in [ISO.IEC.23008-2]) that could be decoded by an appropriate video decoder.

4.2. V3C parameter set

While this memo intends to describe encapsulation of V3C atlas data, aspects related to signaling of V3C parameter set need to be considered. V3C parameter set is signaled in its own V3C unit, which allows decoupling the transmission of V3C parameter set from the V3C video and atlas components. 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. This memo provides information how V3C parameter set can be signaled as part of session description protocol, see Section 10.

4.3. V3C atlas and video components

4.3.1. General

In V3C bitstream the atlas component is identified by vuh_unit_type equal to V3C_AD, or V3C_CAD in 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 and are described in 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 separated 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 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 V3C unit header SHOULD be provided in one form or another to a V3C decoder, e.g. as part of SDP as described in this memo in Section 10. The four-byte V3C unit header syntax and semantics are copied below as defined in [ISO.IEC. 23090-5].

```
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;
  }
  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;
  }
  if( vuh_unit_type == V3C_OVD || vuh_unit_type == V3C_AD ||
      vuh_unit_type == V3C_PVD) {
    bit(17) vuh_reserved_zero_17bits;
  } 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]. As a convenience, the mapping table
from vuh_unit_type values to semantics is copied below 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
4	V3C_AVD	Attribute video data	Attribute information
5	V3C_PVD		Packing information

vuh_unit_type	Identifier	V3C unit type	Description
		Packed video data	
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 indicates if the associated geometry or attribute video data unit is a RAW and/or EOM coded points video only sub-bitstream.

4.3.2. Atlas NAL units

Atlas NAL unit (nal_unit(NumBytesInNalUnit)) is a byte-aligned syntax structure defined by [ISO.IEC.23090-5] to carry atlas data. atlas NAL unit always contains a 16-bit NAL unit header (nal_unit_header()), which indicates among other things the type of the NAL unit (nal_unit_type). The sample code below describes the NAL unit syntax, including the NAL unit header.

```
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 MUST be equal to 0. (F)

nal_unit_type indicates the type of the RBSP data structure
contained in the NAL unit (NUT)

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. (NLI)

nal_temporal_id_plus1 minus 1 indicates a temporal identifier for
the NAL unit. The value of nal_temporal_id_plus1 MUST NOT be equal
to 0. (TID)

4.4. Systems and transport interfaces

In addition to releasing specifications on V3C [ISO.IEC.23090-5] and
[ISO.IEC.23090-12], MPEG is conducting further systems level work on
file format level to encapsulate compressed V3C content. The seventh
edition of the 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
focuses on defining how V3C content SHOULD be stored in a file and
streamed over DASH. 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 SHOULD be considered especially when desiging
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
defintions. Aspects related to RTP header, RTP payload header and
general payload structure are considered along with different
packetization modes.

5.2. RTP Header

The format of the RTP header is specified in [RFC3550] and
replicated below in Figure 1 for convenience. This payload format
uses the fields of the header in a manner consistent with that
specification.

```
 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             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
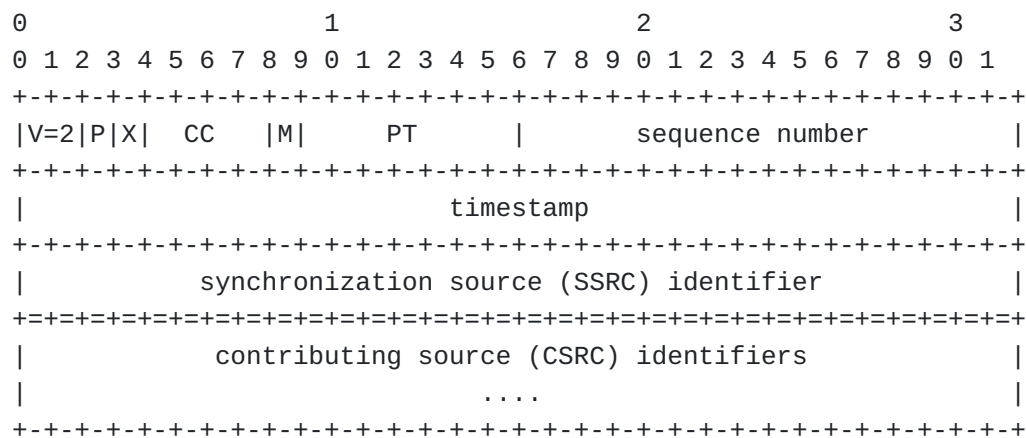
Figure 1: RTP Header

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

Marker bit (M): 1 bit

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

When MRST or MRMT is in use, if an access unit appears in multiple
RTP streams, the marker bit is set on each RTP stream's last packet
of the access unit.

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.

NOTE: (informative) It is not required to use different payload type values for different RTP streams in MRST or MRMT.

Sequence Number (SN): 16 bits

Set and used in accordance with [RFC3550]

Timestamp (32 bits):

The RTP timestamp is set to the sampling timestamp of the content. A 90 kHz clock rate MUST be used.

If the NAL unit has no timing properties of its own (e.g., parameter set and SEI NAL units), the RTP timestamp MUST be set to the RTP timestamp of the coded picture of the access unit in which case 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].

Synchronization source (SSRC): 32 bits

Used to identify the source of the RTP packets.

When using SRST, by definition a single SSRC is used for all parts of a single bitstream. In MRST or MRMT, different SSRCs are used for each RTP stream containing a subset of the sub-layers of the single (temporally scalable) bitstream. A receiver is required to correctly associate the set of SSRCs that are included parts of the same bitstream.

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 described below in Figure 2.

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|F|    NUT    |    NLI    | TID |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
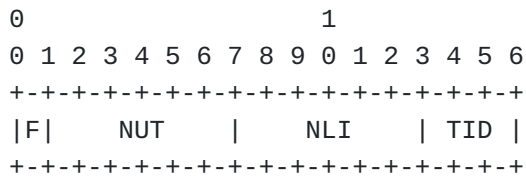
Figure 2: RTP Payload Header

F: nal_forbidden_zero_bit as specified in [ISO.IEC.23090-5] MUST be
equal to 0.

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

NLI: 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: 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. Transmission modes

This memo enables transmission of an V3C atlas bitstream over:

  *a Single RTP stream on a Single media Transport (SRST),

  *Multiple RTP streams over a Single media Transport (MRST), or

  *Multiple RTP streams on Multiple media Transports (MRMT).

When in MRST or MRMT, multiple RTP streams may be grouped together
as specified in [RFC5888] and [RFC9143].

SRST or MRST SHOULD be used for point-to-point unicast scenarios,
whereas MRMT SHOULD be used for point-to-multipoint multicast
scenarios where different receivers require different operation
points of the same V3C atlas bitstream, to improve bandwidth
utilizing efficiency.

NOTE: A multicast may degrade to a unicast at some point when only
one receiver has left. This is a justification of the first "SHOULD"
instead of "MUST". There might be scenarios where MRMT 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".

The transmission mode is indicated by the tx-mode media parameter. If tx-mode is equal to "SRST", SRST MUST be used. Otherwise, if tx-mode is equal to "MRST", MRST MUST be used. Otherwise (tx-mode is equal to "MRMT"), MRMT MUST be used.

NOTE: (informative) When an RTP stream does not depend on other RTP streams, any of SRST, MRST, or MRMT may be in use for the RTP stream.

Receivers MUST support all of SRST, MRST, and MRMT. The required support of MRMT by receivers does not imply that multicast must be supported by receivers.

5.5. Payload structures

5.5.1. 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 5.5.2.

  *Aggregation Packet: Contains multiple NAL units in a single RTP
   payload. This payload structure is specified in Section 5.5.3.

  *Fragmentation Unit: Contains a subset of a single NAL unit. This
   payload structure is specified in Section 5.5.4.

NOTE: (informative) This specification 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.

5.5.2. Single NAL unit packet

Single NAL unit packet contains exactly one NAL unit, and consists of a RTP payload header and following conditional fields: 16-bit DONL and 16-bit v3c-tile-id. The rest of the payload data contain the NAL unit payload data (excluding the NAL unit header). Single NAL unit packet may 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 below in Figure 3.

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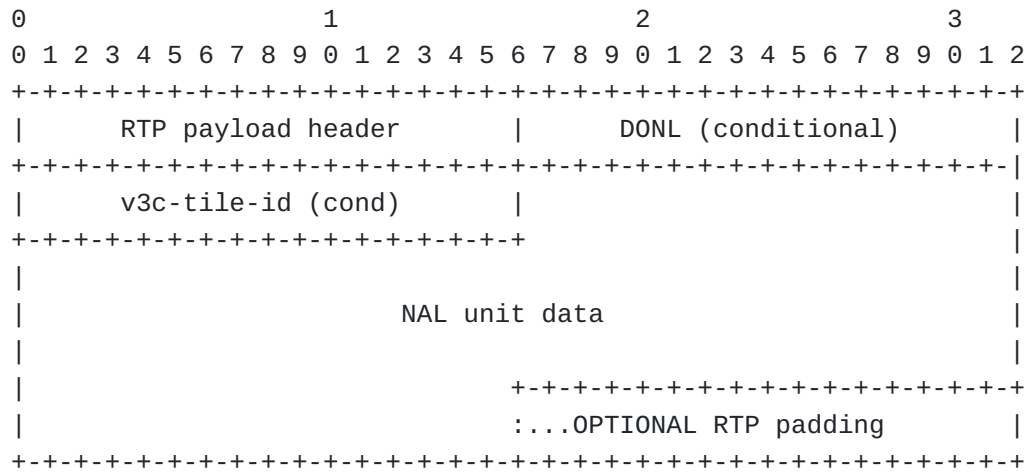
Figure 3: Single NAL unit packet

RTP payload header SHOULD 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. If sprop-max-don-diff is greater than 0 for any of the RTP streams, the DONL field MUST be present, and the variable DONL for the contained NAL unit is derived as equal to the value of the DONL field. Otherwise (sprop-max-don-diff is equal to 0 for all the RTP streams), the DONL field MUST NOT be present.

The v3c-tile-id field, when present, specifies the 16-bit tile identifier for the NAL unit, as signaled in V3C atlas tile header defied in [ISO.IEC.23090-5]. If v3c-tile-id-pres is equal to 1 and 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, defined in [ISO.IEC.23090-5], which means that NAL unit types outside of the range are not specific to atlas tiles and SHOULD NOT contain v3c-tile-ids.

5.5.3. Aggregation packet

Aggregation Packets (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.

Aggregation packets may wrap multiple NAL units belonging to the same access unit in a single RTP payload. The first two bytes of the AP MUST contain 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]. AP may contain a conditional v3c-tile-id field. AP MUST contain two or more aggregation units. The structure of AP 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 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  RTP payload header (NUT=56)  |     v3c-tile-id (cond)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 Two or more aggregation units                 |
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               :...OPTIONAL RTP padding        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
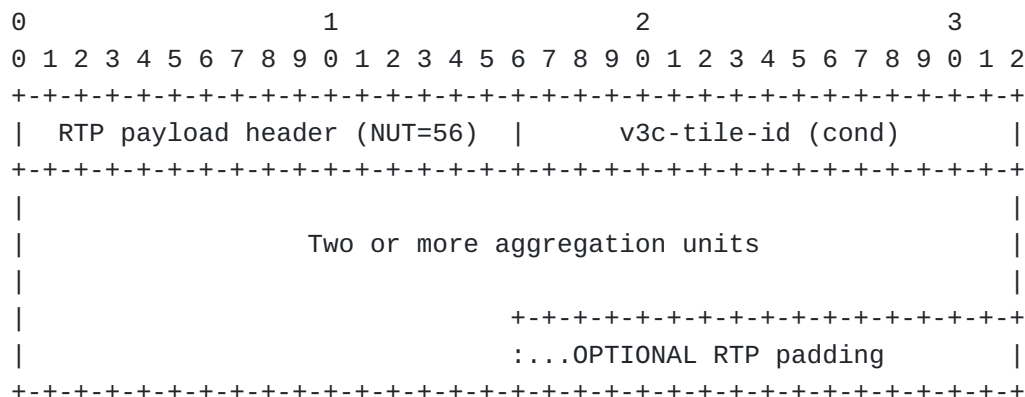
Figure 4: Aggregation Packet (AP)

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 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.

AP MUST carry at least two aggregation units (AU) and can carry as many aggregation units 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 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 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 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  DOND (cond)  /  DONL (cond)  |      v3c-tile-id (cond)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
|            NALU size          |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
|                          NAL unit                             |
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

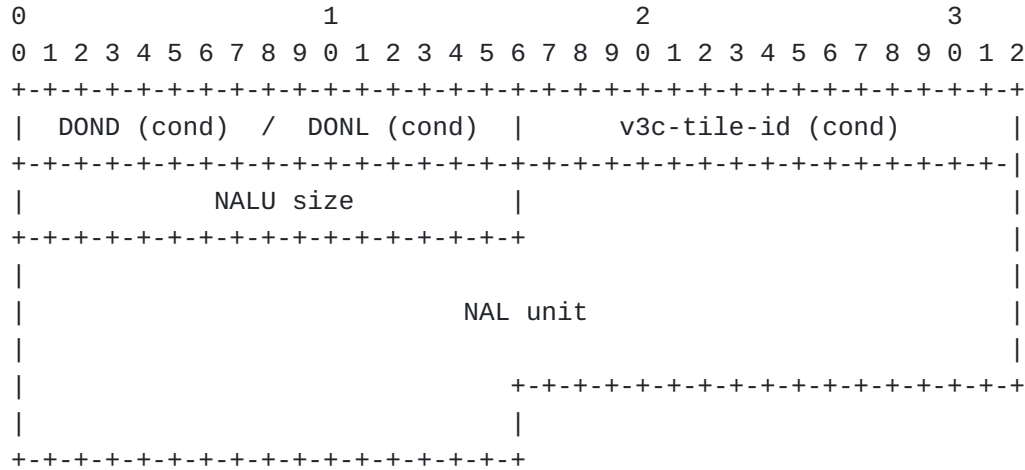   Figure 5: Aggregation Unit (AU)

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 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 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).

5.5.4. Fragmentation unit

Fragmentation Units (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.

A FU consists of a RTP payload header with NUT equal to 58, 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 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 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  RTP payload header (NUT=58)  |   FU header   |  DONL (cond)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
|  DONL (cond)  |    v3c-tile-id (cond)         |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
|                                                               |
|                          FU payload                           |
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               :...OPTIONAL RTP padding        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
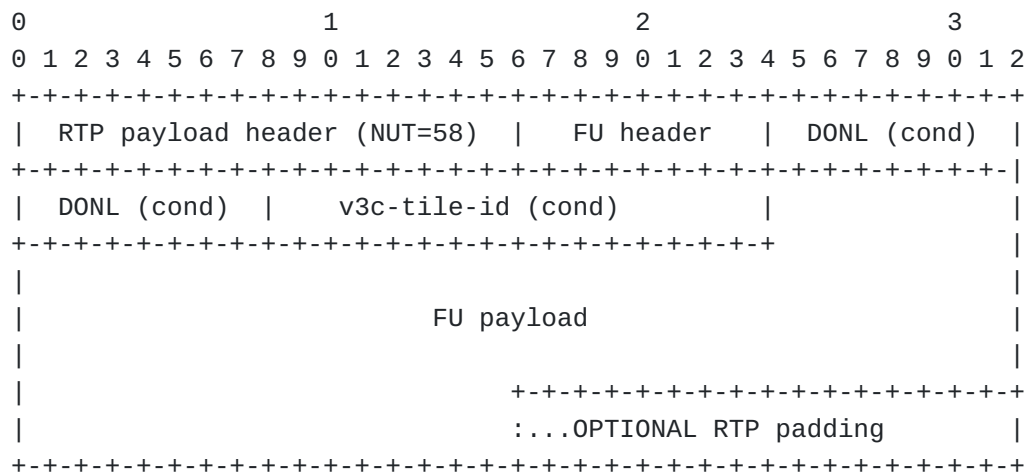
Figure 6: Fragmentation Unit

The fields in the RTP payload header are set as follows. The NUT field MUST be equal to 58. The rest of the fields MUST be equal to the fragmented NAL unit.

The FU header consists of an S bit, an E bit, and a 6-bit FUT field.
The structure of FU header is illustrated below in [Figure 7](#).

```
+---------------+
|0|1|2|3|4|5|6|7|
+-+-+-+-+-+-+-+-+
|S|E|    FUT    |
+-+-+-----------+
```
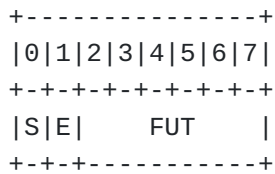
                    Figure 7: Fragmentation unit header

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 field 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 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 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.

5.6. 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 V3C 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 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.

6. Packetization and de-packetization rules

The following packetization rules apply:

*If sprop-max-don-diff is greater than 0 for any of the RTP streams, the transmission order of NAL units carried in the RTP stream MAY be different than the NAL unit decoding order and the NAL unit output order. 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 and, when tx-mode is equal to "MRST" or "MRMT", MUST also be the same as the NAL unit output order.

*A NAL unit of a small size SHOULD be encapsulated in an aggregation packet together with one or more other NAL units in order to avoid 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 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 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 55,
 inclusive, may be passed to the decoder. NAL-unit-like structures
 with NAL unit type values in the range of 55 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].

7. Payload Examples

7.1. General

Examples describing the different payload formats is provided.

7.2. V3C fragmentation unit

This example illustrates how fragmetation unit may be used to divide
one NAL unit into to RTP packets. The Figure 8 illustrates 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=58)  |   FU header   |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
|                                                               |
|                          FU payload                           |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

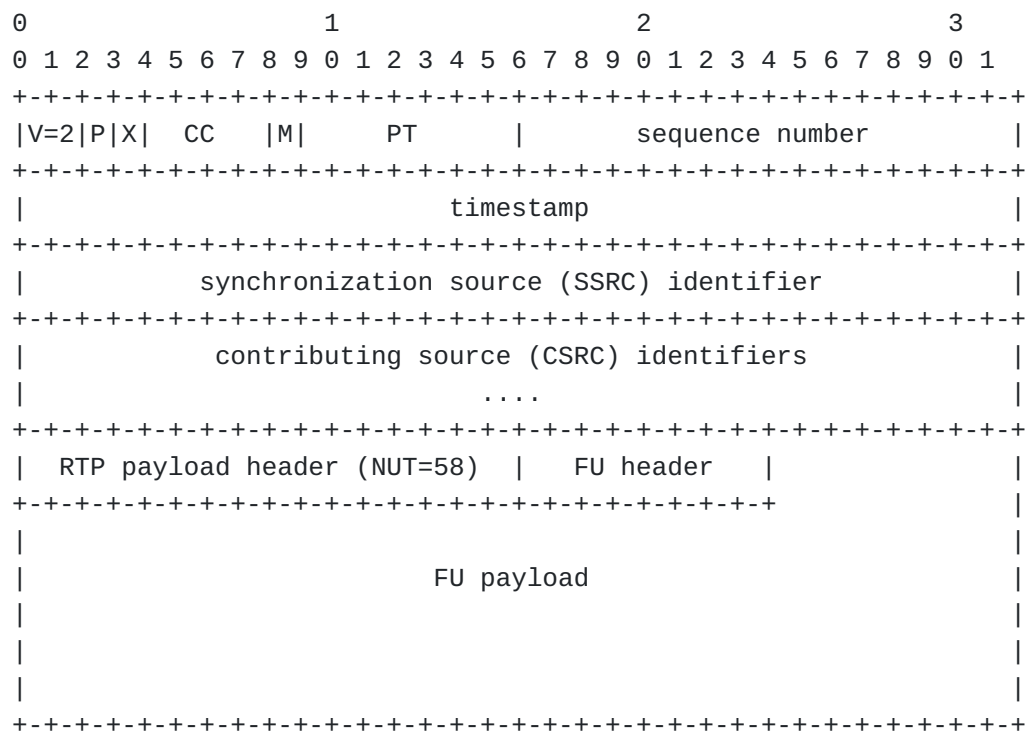   Figure 8: First packet of fragmented NAL unit

The Figure 9 illustrates the structure of the second packet with the
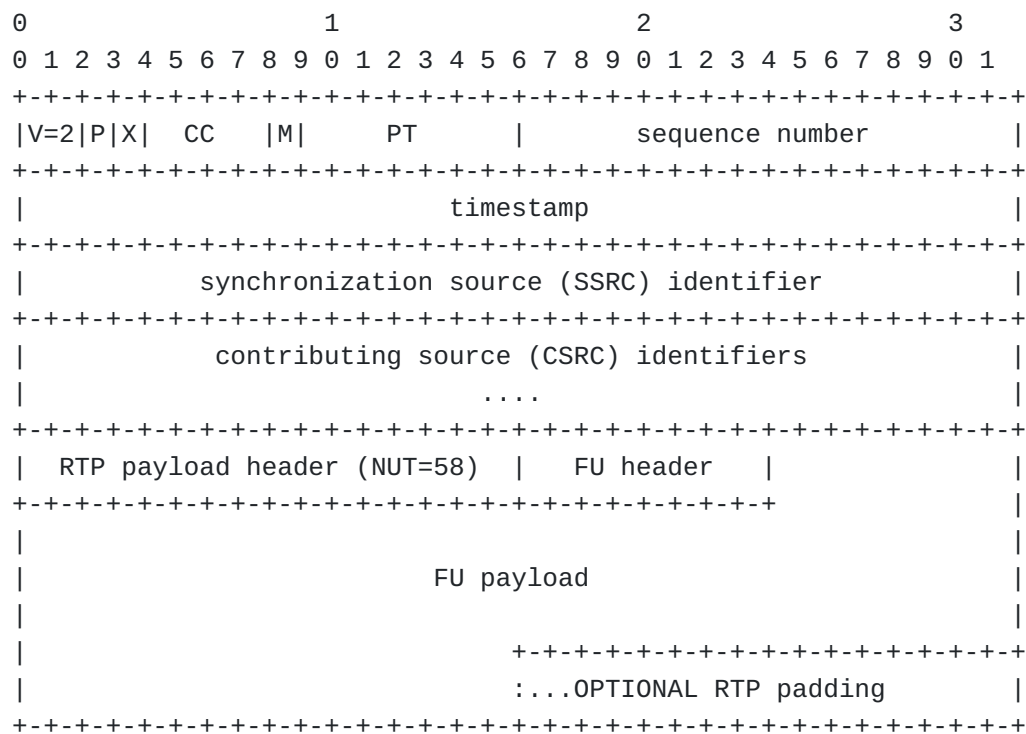rest of the fragmented NAL unit.

```
 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=58)  |   FU header   |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
|                                                               |
|                          FU payload                           |
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               :...OPTIONAL RTP padding         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

             Figure 9: Second packet of fragmented NAL unit
```

## 8.  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 or the RTP stream. The parameters are specified here as part of the media type registration for the V3C codec. 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.

## 8.1.  Media Type Definition

Type name: application

Subtype name: v3c

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

 v3c-unit-header:

provides a V3C unit header bytes defined in [ISO.IEC.23090-5]. The value contains base16 [RFC4648] (hexadecimal) representation of the 4 bytes of V3C unit header.

 v3c-unit-type:

v3c-unit-type provides a V3C unit type value corresponding to vuh_unit_type defined in [ISO.IEC.23090-5], i.e. defines V3C sub-bitstream type.

 v3c-vps-id:

v3c-vps-id provides a value corresponding to vuh_v3c_parameter_set_id defined in [ISO.IEC.23090-5].

 v3c-atlas-id:

v3c-atlas-id provides a value corresponding to vuh_atlas_id defined in [ISO.IEC.23090-5].

 v3c-attr-idx:

v3c-attr-idx provides a value corresponding to vuh_attribute_index defined in [ISO.IEC.23090-5].

v3c-attr-part-idx:

v3c-attr-part-idx provides a value corresponding to
vuh_attribute_partition_index defined in [ISO.IEC.23090-5].

v3c-map-idx:

v3c-map-idx provides a value corresponding to vuh_map_index defined
in [ISO.IEC.23090-5].

v3c-aux-video-flag:

v3c-aux-video-flag provides a value corresponding to
vuh_auxiliary_video_flag defined in [ISO.IEC.23090-5].

v3c-parameter-set:

v3c-parameter-set provides V3C parameter set bytes as defined in
[ISO.IEC.23090-5]. The value contains base16 [RFC4648] (hexadecimal)
representation of the V3C parameter set bytes.

v3c-tile-id:

v3c-tile-id indicates that the RTP stream contains only portion of
the tiles in the atlas. v3c-tile-id is a comma-spearated (',') list
of integer values, which indicate the v3c-tile-ids that are present
in the RTP stream.

v3c-tile-id-pres:

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

v3c-atlas-data:

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 is a
comma-separated (',') list of encoded representations of the atlas
NAL units as specified in [ISO.IEC.23090-5]. The NAL units SHOULD be
encoded as base16 [RFC4648] (hexadecimal) representations.

v3c-common-atlas-data:

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 is a comma-separated (',') list of encoded
representations of the common atlas NAL units (i.e. NAL_CASPS and
NAL_CAF_IDR) as specified in [ISO.IEC.23090-5]. The NAL units SHOULD
be encoded as base16 [RFC4648] (hexadecimal) representations.

v3c-sei:

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 encoded 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]. The SEI NAL units SHOULD be encoded as base16 [RFC4648] (hexadecimal) representations.

v3c-ptl-level-idc:

v3c-ptl-level-idc provides a value corresponding to ptl_level_idc defined in [ISO.IEC.23090-5].

v3c-ptl-tier-flag:

v3c-ptl-tier-flag provides a value corresponding to ptl_tier_flag defined in [ISO.IEC.23090-5].

v3c-ptl-codec-idc:

v3c-ptl-codec-idc provides a value corresponding to ptl_profile_codec_group_idc defined in [ISO.IEC.23090-5].

v3c-ptl-toolset-idc:

v3c-ptl-toolset-idc provides a value corresponding to ptl_profile_toolset_idc defined in [ISO.IEC.23090-5].

v3c-ptl-rec-idc:

v3c-ptl-rec-idc provides a value corresponding to ptl_profile_reconstruction_idc defined in [ISO.IEC.23090-5].

tx-mode:

This parameter indicates whether the transmission mode is SRST, MRST, or MRMT.

The value of tx-mode MUST be equal to "SRST", "MRST" or "MRMT". When not present, the value of tx-mode is inferred to be equal to "SRST".

If the value is equal to "MRST", MRST MUST be in use. Otherwise, if the value is equal to "MRMT", MRMT MUST be in use. Otherwise (the value is equal to "SRST"), SRST MUST be in use.

The value of tx-mode MUST be equal to "MRST" for all RTP streams in an MRST.

The value of tx-mode MUST be equal to "MRMT" for all RTP streams in
an MRMT.

 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.

Encoding considerations:

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

Security considerations:

Please see Section 12.

Interoperability considerations: N/A

Published specification:

Applications that use this media type: N/A

Additional information: N/A

Person & email address to contact for further information:

Intended usage: COMMON

Restrictions on usage: N/A

Author: See Authors' Addresses section of this memo.

Change controller: IETF Payload working group delegated from the
IESG.

Provisional registration? (standards tree only): No

9.  Congestion Control Considerations

This section is to describe the possibility to vary the bitrate as a
response to congestion. Below is also a proposal for an initial text
that reference RTP and profiles definition of congestion control.

Congestion control for RTP SHALL be used in accordance with
[RFC3550], and with any applicable RTP profile: e.g., [RFC3551]. An
additional requirement if best-effort service is being used is users
of this payload format MUST monitor packet loss to ensure that the
packet loss rate is within acceptable parameters.

Circuit Breakers [RFC8083] is an update to RTP [RFC3550] that
defines criteria for when one is required to stop sending RTP Packet
Streams. The circuit breakers is to be implemented and followed.

10.  Session Description Protocol

The mapping of above defined payload format media type is mapped to
fields in the Session Description Protocol (SDP) according to
[RFC8866].

10.1.  Mapping of payload type parameters to SDP

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

An example of media representation in SDP is as follows:

```
m=application 49170 RTP/AVP 98
a=rtpmap:98 v3c/90000
a=fmtp:98 v3c-unit-header=08000000; // V3C_AD
          v3c-ptl-tier-flag=1
```

10.1.2. 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. avc, hevc, vvc etc.

  *The clock rate in the "a=rtpmap" line MUST be 90000.

  *The OPTIONAL parameters v3c-unit-header, v3c-unit-type, v3c-vps-
   id, v3c-atlas-id, v3c-attr-idx, v3c-attr-part-idx, v3c-map-idx,
   v3c-aux-video-flag, sprop-max-don-diff, v3c-parameter-set, v3c-
   atlas-data, v3c-common-atlas-data, v3c-sei, v3c-tile-id, v3c-
   tile-id-pres, v3c-ptl-level-idc, v3c-ptl-tier-flag, v3c-ptl-
   codec-idc, v3c-ptl-toolset-idc, v3c-ptl-rec-idc, 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 may include any optional parameters from
   the respective video payload specifications.

An example of media representation corresponding to occupancy
component in SDP is as follows:

```
 m=video 49170 RTP/AVP 99
 a=rtpmap:99 H265/90000
 a=fmtp:99 sprop-max-don-diff=0;
           v3c-unit-header=10000000
```

When v3c-unit-header or v3c-unit-type indicate V3C unit type
V3C_PVD, v3c-parameter-set, v3c-atlas-data or v3c-common-atlas-data
may be signaled along the video stream. When v3c-parameter-set, v3c-
atlas-data or v3c-common-atlas-data are present it indicates that
the provided data is static for the whole duration of the stream.

When v3c-parameter-set, v3c-atlas-data or v3c-common-atlas-data are
signaled along the video stream it is expected the respective v3c-
parameter-set, v3c-atlas-data or v3c-common-atlas-data remain static
for the duration of the stream.

An example of media representation in SDP is as follows:

```
 m=video 49170 RTP/AVP 99
 a=rtpmap:99 H265/90000
 a=fmtp:99 v3c-unit-header=28000000;
           v3c-parameter-set=F6F0093992;
           v3c-atlas-data=ABCA,5D5A,68
```

10.2. Grouping Framework

Different V3C components can be represented by their own respective
RTP streams. A grouping tool, as defined in [RFC5888], may be
extended to support V3C grouping.

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> <v3c specific session-level parameters>
```

The V3C grouping type attribute related v3c-specific session level
parameters can include the following optional information:

```
 v3c-parameter-set=<value>
 v3c-atlas-data=<value>
 v3c-common-atlas-data=<value>
 v3c-sei=<value>
```

When signaled as a session level parameter, the data is considered
to be static for the duration of the stream.

The following example shows an SDP including four media lines, three
describing V3C video components and one V3C atlas component. All the
media lines are grouped under one V3C group which provides the V3C
parameter set.

```
...
a=group:V3C 1 2 3 4 v3c-parameter-set=AF6F00939921878
m=video 40000 RTP/AVP 96
a=rtpmap:96 H264/90000
a=fmtp:96 v3c-unit-header=10000000 // occupancy
a=mid:1
m=video 40002 RTP/AVP 97
a=rtpmap:97 H264/90000
a=fmtp:97 v3c-unit-header=18000000 // geometry
a=mid:2
m=video 40004 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 v3c-unit-header=20000000 // attribute
a=mid:3
m=application 40008 RTP/AVP 100
a=rtpmap:100 v3c/90000
a=fmtp:100 v3c-unit-header=08000000; // atlas
a=mid:4
```

V3C group attribute type can be used as follows to indicate
different V3C components and associate static atlas data with them.

```
 ...
a=group:v3c 1 2 3 v3c-parameter-set=AF6F00939921878;
                  v3c-atlas-data=ABCA,5D5D,68
m=video 40000 RTP/AVP 96
a=rtpmap:96 H264/90000
a=fmtp:96 v3c-unit-header=10000000; // occupancy
a=mid:1
m=video 40002 RTP/AVP 97
a=rtpmap:97 H264/90000
a=fmtp:96 v3c-unit-header=18000000; // geometry
a=mid:2
m=video 40004 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:96 v3c-unit-header=20000000; // attribute
a=mid:3
```

The following example describes how every v3c video component is
packed in to a single stream and associated with static atlas data.

```
 ...
m=video 40000 RTP/AVP 96
a=rtpmap:96 H265/90000
a=fmtp:96 v3c-unit-header=28000000; // packed video
        v3c-parameter-set=AF6F00939921878;
        v3c-atlas-data=ABCA,5D5D,68
a=mid:1
```

The example below describes how content with two atlases can be
signaled as separate streams.

```
 ...
 a=group:V3C 1 2 3 4 5 6 7 8 v3c-parameter-set=AF6F00939921878;
                              v3c-common-atlas-data=AFFA,0110;
 m=video 40000 RTP/AVP 96
 a=rtpmap:96 H264/90000
 a=fmtp:96 v3c-unit-header=10000000 // occupancy, atlas 0
 a=mid:1
 m=video 40002 RTP/AVP 97
 a=rtpmap:97 H264/90000
 a=fmtp:97 v3c-unit-header=18000000 // geometry, atlas 0
 a=mid:2
 m=video 40004 RTP/AVP 98
 a=rtpmap:98 H264/90000
 a=fmtp:98 v3c-unit-header=20000000 // attribute, atlas 0
 a=mid:3
 m=application 40008 RTP/AVP 100
 a=rtpmap:100 v3c/90000
 a=fmtp:100 v3c-unit-header=08000000; // atlas 0
 a=mid:4
 m=video 40010 RTP/AVP 101
 a=rtpmap:101 H264/90000
 a=fmtp:101 v3c-unit-header=10020000 // occupancy, atlas 1
 a=mid:5
 m=video 40012 RTP/AVP 102
 a=rtpmap:102 H264/90000
 a=fmtp:102 v3c-unit-header=18020000 // geometry, atlas 1
 a=mid:6
 m=video 40014 RTP/AVP 103
 a=rtpmap:103 H264/90000
 a=fmtp:103 v3c-unit-header=20020000 // attribute, atlas 1
 a=mid:7
 m=application 40018 RTP/AVP 104
 a=rtpmap:104 v3c/90000
 a=fmtp:104 v3c-unit-header=08020000; // V3C_AD, atlas 1
 a=mid:8
```

10.3. Offer/Answer Considerations

An example of offer which only sends V3C content. The following
example contains video components at three different versions.

```
...
a=group:v3c 1 2 3 4 v3c-ptl-level-idc=10;
                    v3c-parameter-set=AF6F00939921878
m=video 40000 RTP/AVP 96 97 98
a=rtpmap:96 H264/90000
a=rtpmap:97 H265/90000
a=rtpmap:98 H266/90000
a=fmtp:96 v3c-unit-type=2;v3c-vps-id=0;v3c-atlas-id=0
a=fmtp:97 v3c-unit-type=2;v3c-vps-id=0;v3c-atlas-id=0
a=fmtp:98 v3c-unit-type=2;v3c-vps-id=0;v3c-atlas-id=0
a=sendonly
a=mid:1
m=video 40002 RTP/AVP 96 97 98
a=rtpmap:96 H264/90000
a=rtpmap:97 H265/90000
a=rtpmap:98 H266/90000
a=fmtp:96 v3c-unit-type=3;v3c-vps-id=0;v3c-atlas-id=0;
a=fmtp:97 v3c-unit-type=3;v3c-vps-id=0;v3c-atlas-id=0;
a=fmtp:98 v3c-unit-type=3;v3c-vps-id=0;v3c-atlas-id=0;
a=mid:2
a=sendonly
m=video 40004 RTP/AVP 96 97 98
a=rtpmap:96 H264/90000
a=rtpmap:97 H265/90000
a=rtpmap:98 H266/90000
a=fmtp:96 v3c-unit-type=4;v3c-vps-id=0;v3c-atlas-id=0
a=fmtp:97 v3c-unit-type=4;v3c-vps-id=0;v3c-atlas-id=0
a=fmtp:98 v3c-unit-type=4;v3c-vps-id=0;v3c-atlas-id=0
a=mid:3
a=sendonly
m=application 40006 RTP/AVP 100
a=rtpmap:100 v3c/90000
a=fmtp:100 v3c-unit-type=1;v3c-vps-id=0;v3c-atlas-id=0
a=mid:4
a=sendonly
```

An example of answer which 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
m=video 50002 RTP/AVP 97
a=rtpmap:97 H265/90000
a=recvonly
m=video 50004 RTP/AVP 98
a=rtpmap:98 H266/90000
a=recvonly
m=application 50006 RTP/AVP 96
a=rtpmap:96 v3c/90000
a=recvonly
```

An example offer, which allows bundling different V3C components on one stream, based on [RFC9143].

```
 ...
 a=group:BUNDLE 1 2 3 4
 a=group:v3c 1 2 3 4 v3c-parameter-set=AF6F00939921878
 m=video 40000 RTP/AVP 96
 a=rtpmap:96 H264/90000
 a=fmtp:96 v3c-unit-type=2;v3c-vps-id=0;v3c-atlas-id=0
 a=mid:1
 a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
 m=video 40002 RTP/AVP 96
 a=rtpmap:96 H264/90000
 a=fmtp:96 v3c-unit-type=3;v3c-vps-id=0;v3c-atlas-id=0;
 a=mid:2
 a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
 m=video 40004 RTP/AVP 96
 a=rtpmap:96 H264/90000
 a=fmtp:96 v3c-unit-type=4;v3c-vps-id=0;v3c-atlas-id=0
 a=mid:3
 a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
 m=application 40006 RTP/AVP 97
 a=rtpmap:97 v3c/90000
 a=fmtp:97 v3c-unit-type=1;v3c-vps-id=0;v3c-atlas-id=0
 a=mid:4
 a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
```

An example answer, which accepts 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 96
a=rtpmap:96 H264/90000
a=bundle-only
a=mid:2
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 0 RTP/AVP 96
a=rtpmap:96 H264/90000
a=bundle-only
a=mid:3
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
m=application 0 RTP/AVP 97
a=rtpmap:97 v3c/90000
a=bundle-only
a=mid:4
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
```

10.4. Declarative SDP Considerations

Placeholder

11. IANA Considerations

Placeholder

12. Security Considerations

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

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.

13. References

13.1. Normative References

[ISO.IEC.23090-12] ISO/IEC, "Information technology --- Coded representation of immersive media --- Part 12: MPEG Immersive video (MIV)", ISO/IEC 23090-12, 2022, <https://www.iso.org/standard/79113.html>.

[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, 2021, <https://www.iso.org/standard/73025.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>.

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

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

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

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

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

[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>.

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

[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", DOI 10.17487/RFC6190, RFC 6190, May 2011,
             <https://www.rfc-editor.org/info/rfc6190>.

[RFC6838]   Freed, N., Klensin, J., and T. Hansen, "Media Type
             Specifications and Registration Procedures", BCP 13, DOI
             10.17487/RFC6838, RFC 6838, January 2013, <https://
             www.rfc-editor.org/info/rfc6838>.

[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>.

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

[RFC8866]   Begen, A., Kyzivat, P., Perkins, C., and M. Handley,
             "SDP: Session Description Protocol", DOI 10.17487/
             RFC8866, RFC 8866, 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>.

13.2. Informative References

[ISO.IEC.14496-10]

ISO/IEC, "Information technology - Coding of audio-visual objects - Part 10: Advanced video coding", ISO/IEC 14496-10, 2020, <https://www.iso.org/standard/75400.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, 2020, <https://www.iso.org/standard/74428.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, 2020, <https://www.iso.org/standard/75484.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 FDIS 23090-10, 2022, <https://www.iso.org/standard/78991.html>.

[ISO.IEC.23090-3] ISO/IEC, "Information technology --- Coded representation of immersive media --- Part 3: Versatile video coding", ISO/IEC 23090-3, 2021, <https://www.iso.org/standard/73022.html>.

[RFC7201]   Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", DOI 10.17487/RFC7201, RFC 7201, 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>.

Authors' Addresses

Lauri Ilola
Nokia Technologies
Werinherstrasse 91
D-81541 Munich
Germany

Email: lauri.ilola@nokia.com

Lukasz Kondrad
Nokia Technologies
Werinherstrasse 91
D-81541 Munich

Germany

Email: lukasz.kondrad@nokia.com