Internet Engineering Task Force (IETF)
Request for Comments: 6184
Obsoletes: 3984
Category: Standards Track
ISSN: 2070-1721

Y.-K. Wang
R. Even
Huawei Technologies
T. Kristensen
Tandberg
R. Jesup
WorldGate Communications

May 2011

RTP Payload Format for H.264 Video

Abstract

This memo describes an RTP Payload format for the ITU-T Recommendation H.264 video codec and the technically identical ISO/IEC International Standard 14496-10 video codec, excluding the Scalable Video Coding (SVC) extension and the Multiview Video Coding extension, for which the RTP payload formats are defined elsewhere. The RTP payload format allows for packetization of one or more Network Abstraction Layer Units (NALUs), produced by an H.264 video encoder, in each RTP payload. The payload format has wide applicability, as it supports applications from simple low bitrate conversational usage, to Internet video streaming with interleaved transmission, to high bitrate video-on-demand.

This memo obsoletes RFC 3984. Changes from RFC 3984 are summarized in Section 14. Issues on backward compatibility to RFC 3984 are discussed in Section 15.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6184.

Table of Contents

1. Introduction

This memo specifies an RTP payload specification for the video coding standard known as ITU-T Recommendation H.264 [1] and ISO/IEC International Standard 14496-10 [2] (both also known as Advanced Video Coding (AVC)). In this memo, the name H.264 is used for the codec and the standard, but this memo is equally applicable to the ISO/IEC counterpart of the coding standard.

This memo obsoletes RFC 3984. Changes from RFC 3984 are summarized in Section 14. Issues on backward compatibility to RFC 3984 are discussed in Section 15.

1.1. The H.264 Codec

The H.264 video codec has a very broad application range that covers all forms of digital compressed video, from low bitrate Internet streaming applications to HDTV broadcast and Digital Cinema applications with nearly lossless coding. Compared to the current state of technology, the overall performance of H.264 is such that bitrate savings of 50% or more are reported. Digital Satellite TV quality, for example, was reported to be achievable at 1.5 Mbit/s, compared to the current operation point of MPEG 2 video at around 3.5 Mbit/s [10].

The codec specification [1] itself conceptually distinguishes between a Video Coding Layer (VCL) and a Network Abstraction Layer (NAL). The VCL contains the signal processing functionality of the codec; mechanisms such as transform, quantization, and motion-compensated prediction; and a loop filter. It follows the general concept of most of today's video codecs, a macroblock-based coder that uses inter picture prediction with motion compensation and transform coding of the residual signal. The VCL encoder outputs slices: a bit string that contains the macroblock data of an integer number of macroblocks and the information of the slice header (containing the spatial address of the first macroblock in the slice, the initial quantization parameter, and similar information). Macroblocks in slices are arranged in scan order unless a different macroblock allocation is specified using the syntax of slice groups. In-picture prediction is used only within a slice. More information is provided in [10].

The NAL encoder encapsulates the slice output of the VCL encoder into Network Abstraction Layer Units (NALUs), which are suitable for transmission over packet networks or for use in packet-oriented

multiplex environments. Annex B of H.264 defines an encapsulation process to transmit such NALUs over bytestream-oriented networks. In the scope of this memo, Annex B is not relevant.

Internally, the NAL uses NAL units. A NAL unit consists of a one-byte header and the payload byte string. The header indicates the type of the NAL unit, the (potential) presence of bit errors or syntax violations in the NAL unit payload, and information regarding the relative importance of the NAL unit for the decoding process. This RTP payload specification is designed to be unaware of the bit string in the NAL unit payload.

One of the main properties of H.264 is the complete decoupling of the transmission time, the decoding time, and the sampling or presentation time of slices and pictures. The decoding process specified in H.264 is unaware of time, and the H.264 syntax does not carry information such as the number of skipped frames (as is common in the form of the Temporal Reference in earlier video compression standards). Also, there are NAL units that affect many pictures and that are, therefore, inherently timeless. For this reason, the handling of the RTP timestamp requires some special considerations for NAL units for which the sampling or presentation time is not defined or, at transmission time, is unknown.

1.2. Parameter Set Concept

One very fundamental design concept of H.264 is to generate self-contained packets, to make mechanisms such as the header duplication of RFC 4629 [11] or MPEG-4 Visual's Header Extension Code (HEC) [12] unnecessary. This was achieved by decoupling information relevant to more than one slice from the media stream. This higher-layer meta information should be sent reliably, asynchronously, and in advance from the RTP packet stream that contains the slice packets. (Provisions for sending this information in-band are also available for applications that do not have an out-of-band transport channel appropriate for the purpose). The combination of the higher-level parameters is called a parameter set. The H.264 specification includes two types of parameter sets: sequence parameter sets and picture parameter sets. An active sequence parameter set remains unchanged throughout a coded video sequence, and an active picture parameter set remains unchanged within a coded picture. The sequence and picture parameter set structures contain information such as picture size, optional coding modes employed, and macroblock to slice group map.

To be able to change picture parameters (such as the picture size) without having to transmit parameter set updates synchronously to the slice packet stream, the encoder and decoder can maintain a list of more than one sequence and picture parameter set. Each slice header contains a codeword that indicates the sequence and picture parameter set to be used.

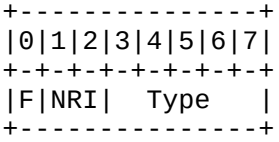
This mechanism allows the decoupling of the transmission of parameter sets from the packet stream and the transmission of them by external means (e.g., as a side effect of the capability exchange) or through a (reliable or unreliable) control protocol. It may even be possible that they are never transmitted but are fixed by an application design specification.

1.3. Network Abstraction Layer Unit Types

Tutorial information on the NAL design can be found in [13], [14], and [15].

All NAL units consist of a single NAL unit type octet, which also co-serves as the payload header of this RTP payload format. A description of the payload of a NAL unit follows.

The syntax and semantics of the NAL unit type octet are specified in [1], but the essential properties of the NAL unit type octet are summarized below. The NAL unit type octet has the following format:



The semantics of the components of the NAL unit type octet, as specified in the H.264 specification, are described briefly below.

F: 1 bit
forbidden_zero_bit. The H.264 specification declares a value of 1 as a syntax violation.

NRI: 2 bits
nal_ref_idc. A value of 00 indicates that the content of the NAL unit is not used to reconstruct reference pictures for inter picture prediction. Such NAL units can be discarded without risking the integrity of the reference pictures. Values greater than 00 indicate that the decoding of the NAL unit is required to maintain the integrity of the reference pictures.

Type: 5 bits
nal_unit_type. This component specifies the NAL unit payload type as defined in Table 7-1 of [1] and later within this memo. For a reference of all currently defined NAL unit types and their semantics, please refer to Section 7.4.1 in [1].

This memo introduces new NAL unit types, which are presented in Section 5.2. The NAL unit types defined in this memo are marked as unspecified in [1]. Moreover, this specification extends the semantics of F and NRI as described in Section 5.3.

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 RFC 2119 [4].

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

3. Scope

This payload specification can only be used to carry the "naked" H.264 NAL unit stream over RTP and not the bitstream format discussed in Annex B of H.264. Likely, the first applications of this specification will be in the conversational multimedia field, video telephony or video conferencing, but the payload format also covers other applications, such as Internet streaming and TV over IP.

4. Definitions and Abbreviations

4.1. Definitions

This document uses the definitions of [1]. The following terms, defined in [1], are summed up for convenience:

access unit: A set of NAL units always containing a primary coded picture. In addition to the primary coded picture, an access unit may also contain one or more redundant coded pictures or other NAL units not containing slices or slice data partitions of a coded picture. The decoding of an access unit always results in a decoded picture.

coded video sequence: A sequence of access units that consists, in decoding order, of an instantaneous decoding refresh (IDR) access unit followed by zero or more non-IDR access units including all subsequent access units up to but not including any subsequent IDR access unit.

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

IDR picture: A coded picture containing only slices with I or SI slice types that causes a "reset" in the decoding process. After the decoding of an IDR picture, all following coded pictures in decoding order can be decoded without inter prediction from any picture decoded prior to the IDR picture.

primary coded picture: The coded representation of a picture to be used by the decoding process for a bitstream conforming to H.264. The primary coded picture contains all macroblocks of the picture.

redundant coded picture: A coded representation of a picture or a part of a picture. The content of a redundant coded picture shall not be used by the decoding process for a bitstream conforming to H.264. The content of a redundant coded picture may be used by the decoding process for a bitstream that contains errors or losses.

VCL NAL unit: A collective term used to refer to coded slice and coded data partition NAL units.

In addition, the following definitions apply:

decoding order number (DON): A field in the payload structure or a derived variable indicating NAL unit decoding order. Values of DON are in the range of 0 to 65535, inclusive. After reaching the maximum value, the value of DON wraps around to 0.

NAL unit decoding order: A NAL unit order that conforms to the constraints on NAL unit order given in Section 7.4.1.2 in [1].

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

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

media-aware network element (MANE): A network element, such as a middlebox or application layer gateway that is capable of parsing certain aspects of the RTP payload headers or the RTP payload and reacting to the contents.

Informative note: The concept of a MANE goes beyond normal routers or gateways in that a MANE has to be aware of the signaling (e.g., to learn about the payload type mappings of the media streams) and that it has to be trusted when working with Secure Real-time Transport Protocol (SRTP). The advantage of using MANEs is that they allow packets to be dropped according to the needs of the media coding. For example, if a MANE has to drop packets due to congestion on a certain link, it can identify and remove those packets whose elimination produces the least adverse effect on the user experience.

static macroblock: A certain amount of macroblocks in the video stream can be defined as static, as defined in Section 8.3.2.8 in [3]. Static macroblocks free up additional processing cycles for the handling of non-static macroblocks. Based on a given amount of video processing resources and a given resolution, a higher number of static macroblocks enables a correspondingly higher frame rate.

default sub-profile: The subset of coding tools, which may be all coding tools of one profile or the common subset of coding tools of more than one profile, indicated by the profile-level-id parameter.

default level: The level indicated by the profile-level-id parameter, which consists of three octets, profile_idc, profile-iop, and level_idc. The default level is indicated by level_idc in most cases, and, in some cases, additionally by profile-iop.

4.2. Abbreviations

DON:	Decoding Order Number
DONB:	Decoding Order Number Base
DOND:	Decoding Order Number Difference
FEC:	Forward Error Correction
FU:	Fragmentation Unit
IDR:	Instantaneous Decoding Refresh
IEC:	International Electrotechnical Commission
ISO:	International Organization for Standardization
ITU-T:	International Telecommunication Union, Telecommunication Standardization Sector
MANE:	Media-Aware Network Element
MTAP:	Multi-Time Aggregation Packet

MTAP16:	MTAP with 16-bit timestamp offset
MTAP24:	MTAP with 24-bit timestamp offset
NAL:	Network Abstraction Layer
NALU:	NAL Unit
SAR:	Sample Aspect Ratio
SEI:	Supplemental Enhancement Information
STAP:	Single-Time Aggregation Packet
STAP-A:	STAP type A
STAP-B:	STAP type B
TS:	Timestamp
VCL:	Video Coding Layer
VUI:	Video Usability Information

5. RTP Payload Format

5.1. RTP Header Usage

The format of the RTP header is specified in RFC 3550 [5] and reprinted in Figure 1 for convenience. This payload format uses the fields of the header in a manner consistent with that specification.

When one NAL unit is encapsulated per RTP packet, the RECOMMENDED RTP payload format is specified in Section 5.6. The RTP payload (and the settings for some RTP header bits) for aggregation packets and fragmentation units are specified in Sections 5.7.2 and 5.8, respectively.

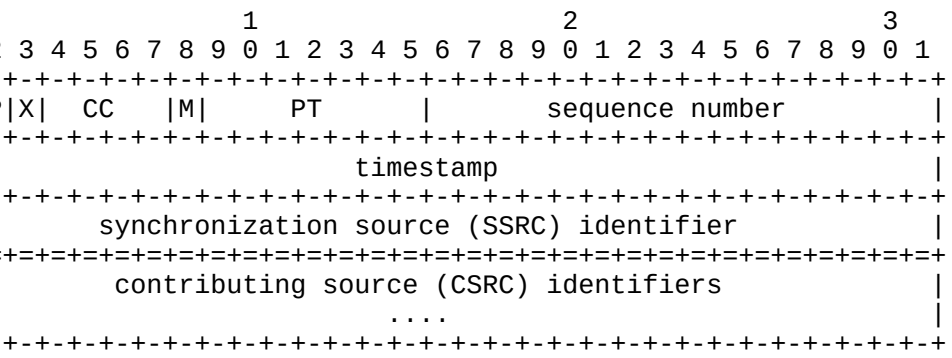


Figure 1. RTP header according to RFC 3550

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

Marker bit (M): 1 bit
Set for the very last packet of the access unit indicated by the RTP timestamp, in line with the normal use of the M bit in video

formats, to allow an efficient playout buffer handling. For aggregation packets (STAP and MTAP), the marker bit in the RTP header MUST be set to the value that the marker bit of the last NAL unit of the aggregation packet would have been if it were transported in its own RTP packet. Decoders MAY use this bit as an early indication of the last packet of an access unit but MUST NOT rely on this property.

Informative note: Only one M bit is associated with an aggregation packet carrying multiple NAL units. Thus, if a gateway has re-packetized an aggregation packet into several packets, it cannot reliably set the M bit of those packets.

Payload type (PT): 7 bits
The assignment of an RTP payload type for this new packet format is outside the scope of this document and will not be specified here. The assignment of a payload type has to be performed either through the profile used or in a dynamic way.

Sequence number (SN): 16 bits
Set and used in accordance with RFC 3550. For the single NALU and non-interleaved packetization mode, the sequence number is used to determine decoding order for the NALU.

Timestamp: 32 bits
The RTP timestamp is set to the sampling timestamp of the content. A 90 kHz clock rate MUST be used.

If the NAL unit has no timing properties of its own (e.g., parameter set and SEI NAL units), the RTP timestamp is set to the RTP timestamp of the primary coded picture of the access unit in which the NAL unit is included, according to Section 7.4.1.2 of [1].

The setting of the RTP timestamp for MTAPs is defined in Section 5.7.2.

Receivers SHOULD ignore any picture timing SEI messages included in access units that have only one display timestamp. Instead, receivers SHOULD use the RTP timestamp for synchronizing the display process.

If one access unit has more than one display timestamp carried in a picture timing SEI message, then the information in the SEI message SHOULD be treated as relative to the RTP timestamp, with the earliest event occurring at the time given by the RTP timestamp and subsequent events later, as given by the difference in picture time values carried in the picture timing SEI message.

Let tSEI1, tSEI2, ..., tSEIn be the display timestamps carried in the SEI message of an access unit, where tSEI1 is the earliest of all such timestamps. Let tmadjst() be a function that adjusts the SEI messages time scale to a 90-kHz time scale. Let TS be the RTP timestamp. Then, the display time for the event associated with tSEI1 is TS. The display time for the event with tSEIx, where x is [2..n], is TS + tmadjst (tSEIx - tSEI1).

Informative note: Displaying coded frames as fields is needed commonly in an operation known as 3:2 pulldown, in which film content that consists of coded frames is displayed on a display using interlaced scanning. The picture timing SEI message enables carriage of multiple timestamps for the same coded picture, and therefore the 3:2 pulldown process is perfectly controlled. The picture timing SEI message mechanism is necessary because only one timestamp per coded frame can be conveyed in the RTP timestamp.

5.2. Payload Structures

The payload format defines three different basic payload structures. A receiver can identify the payload structure by the first byte of the RTP packet payload, which co-serves as the RTP payload header and, in some cases, as the first byte of the payload. This byte is always structured as a NAL unit header. The NAL unit type field indicates which structure is present. The possible structures are as follows.

Single NAL Unit Packet: Contains only a single NAL unit in the payload. The NAL header type field is equal to the original NAL unit type, i.e., in the range of 1 to 23, inclusive. Specified in Section 5.6.

Aggregation Packet: Packet type used to aggregate multiple NAL units into a single RTP payload. This packet exists in four versions, the Single-Time Aggregation Packet type A (STAP-A), the Single-Time Aggregation Packet type B (STAP-B), Multi-Time Aggregation Packet (MTAP) with 16-bit offset (MTAP16), and Multi-Time Aggregation Packet (MTAP) with 24-bit offset (MTAP24). The NAL unit type numbers assigned for STAP-A, STAP-B, MTAP16, and MTAP24 are 24, 25, 26, and 27, respectively. Specified in Section 5.7.

Fragmentation Unit: Used to fragment a single NAL unit over multiple RTP packets. Exists with two versions, FU-A and FU-B, identified with the NAL unit type numbers 28 and 29, respectively. Specified in Section 5.8.

Informative note: This specification does not limit the size of NAL units encapsulated in single NAL unit packets and fragmentation units. The maximum size of a NAL unit encapsulated in any aggregation packet is 65535 bytes.

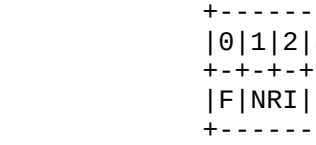
Table 1 summarizes NAL unit types and the corresponding RTP packet types when each of these NAL units is directly used as a packet payload, and where the types are described in this memo.

Table 1. Summary of NAL unit types and the corresponding packet types

NAL Unit Type	Packet Type	Packet Type Name	Section
0	reserved		-
1-23	NAL unit	Single NAL unit packet	5.6
24	STAP-A	Single-time aggregation packet	5.7.1
25	STAP-B	Single-time aggregation packet	5.7.1
26	MTAP16	Multi-time aggregation packet	5.7.2
27	MTAP24	Multi-time aggregation packet	5.7.2
28	FU-A	Fragmentation unit	5.8
29	FU-B	Fragmentation unit	5.8
30-31	reserved		-

5.3. NAL Unit Header Usage

The structure and semantics of the NAL unit header were introduced in Section 1.3. For convenience, the format of the NAL unit header is reprinted below:



This section specifies the semantics of F and NRI according to this specification.

F: 1 bit
forbidden_zero_bit. A value of 0 indicates that the NAL unit type octet and payload should not contain bit errors or other syntax violations. A value of 1 indicates that the NAL unit type octet and payload may contain bit errors or other syntax violations.

MANEs SHOULD set the F bit to indicate detected bit errors in the NAL unit. The H.264 specification requires that the F bit be equal to 0. When the F bit is set, the decoder is advised that bit errors or any other syntax violations may be present in the payload or in the NAL unit type octet. The simplest decoder reaction to a NAL unit in which the F bit is equal to 1 is to discard such a NAL unit and to conceal the lost data in the discarded NAL unit.

NRI: 2 bits
nal_ref_idc. The semantics of value 00 and a non-zero value remain unchanged from the H.264 specification. In other words, a value of 00 indicates that the content of the NAL unit is not used to reconstruct reference pictures for inter picture prediction. Such NAL units can be discarded without risking the integrity of the reference pictures. Values greater than 00 indicate that the decoding of the NAL unit is required to maintain the integrity of the reference pictures.

In addition to the specification above, according to this RTP payload specification, values of NRI indicate the relative transport priority, as determined by the encoder. MANEs can use this information to protect more important NAL units better than they do less important NAL units. The highest transport priority is 11, followed by 10, and then by 01; finally, 00 is the lowest.

> Informative note: Any non-zero value of NRI is handled identically in H.264 decoders. Therefore, receivers need not manipulate the value of NRI when passing NAL units to the decoder.

An H.264 encoder MUST set the value of NRI according to the H.264 specification (Subclause 7.4.1) when the value of nal_unit_type is in the range of 1 to 12, inclusive. In particular, the H.264 specification requires that the value of NRI SHALL be equal to 0 for all NAL units having nal_unit_type equal to 6, 9, 10, 11, or 12.

For NAL units having nal_unit_type equal to 7 or 8 (indicating a sequence parameter set or a picture parameter set, respectively), an H.264 encoder SHOULD set the value of NRI to 11 (in binary format). For coded slice NAL units of a primary coded picture having nal_unit_type equal to 5 (indicating a coded slice belonging to an IDR picture), an H.264 encoder SHOULD set the value of NRI to 11 (in binary format).

For a mapping of the remaining nal_unit_types to NRI values, the following example MAY be used and has been shown to be efficient in a certain environment [14]. Other mappings MAY also be desirable, depending on the application and the H.264 profile in use.

Informative note: Data partitioning is not available in certain profiles, e.g., in the Main or Baseline profiles. Consequently, the NAL unit types 2, 3, and 4 can occur only if the video bitstream conforms to a profile in which data partitioning is allowed and not in streams that conform to the Main or Baseline profiles.

Table 2. Example of NRI values for coded slices and coded slice data partitions of primary coded reference pictures

NAL Unit Type	Content of NAL Unit	NRI (binary)
1	non-IDR coded slice	10
2	Coded slice data partition A	10
3	Coded slice data partition B	01
4	Coded slice data partition C	01

Informative note: As mentioned before, the NRI value of non-reference pictures is 00 as mandated by H.264.

An H.264 encoder SHOULD set the value of NRI for coded slice and coded slice data partition NAL units of redundant coded reference pictures equal to 01 (in binary format).

Definitions of the values for NRI for NAL unit types 24 to 29, inclusive, are given in Sections 5.7 and 5.8 of this memo.

No recommendation for the value of NRI is given for NAL units having nal_unit_type in the range of 13 to 23, inclusive, because these values are reserved for ITU-T and ISO/IEC. No recommendation for the value of NRI is given for NAL units having nal_unit_type equal to 0 or in the range of 30 to 31, inclusive, as the semantics of these values are not specified in this memo.

5.4. Packetization Modes

This memo specifies three cases of packetization modes:

- o Single NAL unit mode
- o Non-interleaved mode
- o Interleaved mode

The single NAL unit mode is targeted for conversational systems that comply with ITU-T Recommendation H.241 [3] (see Section 12.1). The non-interleaved mode is targeted for conversational systems that may not comply with ITU-T Recommendation H.241. In the non-interleaved mode, NAL units are transmitted in NAL unit decoding order. The interleaved mode is targeted for systems that do not require very low end-to-end latency. The interleaved mode allows transmission of NAL units out of NAL unit decoding order.

The packetization mode in use MAY be signaled by the value of the OPTIONAL packetization-mode media type parameter. The used packetization mode governs which NAL unit types are allowed in RTP payloads. Table 3 summarizes the allowed packet payload types for each packetization mode. Packetization modes are explained in more detail in Section 6.

Table 3. Summary of allowed NAL unit types for each packetization mode (yes = allowed, no = disallowed, ig = ignore)

Payload Type	Packet Type	Single NAL Unit Mode	Non-Interleaved Mode	Interleaved Mode
0	reserved	ig	ig	ig
1-23	NAL unit	yes	yes	no
24	STAP-A	no	yes	no
25	STAP-B	no	no	yes
26	MTAP16	no	no	yes
27	MTAP24	no	no	yes
28	FU-A	no	yes	yes
29	FU-B	no	no	yes
30-31	reserved	ig	ig	ig

Some NAL unit or payload type values (indicated as reserved in Table 3) are reserved for future extensions. NAL units of those types SHOULD NOT be sent by a sender (direct as packet payloads, as aggregation units in aggregation packets, or as fragmented units in FU packets) and MUST be ignored by a receiver. For example, the payload types 1-23, with the associated packet type "NAL unit", are

allowed in "Single NAL Unit Mode" and in "Non-Interleaved Mode" but disallowed in "Interleaved Mode". However, NAL units of NAL unit types 1-23 can be used in "Interleaved Mode" as aggregation units in STAP-B, MTAP16, and MTAP24 packets as well as fragmented units in FU-A and FU-B packets. Similarly, NAL units of NAL unit types 1-23 can also be used in the "Non-Interleaved Mode" as aggregation units in STAP-A packets or fragmented units in FU-A packets, in addition to being directly used as packet payloads.

5.5. Decoding Order Number (DON)

In the interleaved packetization mode, the transmission order of NAL units is allowed to differ from the decoding order of the NAL units. Decoding order number (DON) is a field in the payload structure or a derived variable that indicates the NAL unit decoding order. Rationale and examples of use cases for transmission out of decoding order and for the use of DON are given in Section 13.

The coupling of transmission and decoding order is controlled by the OPTIONAL sprop-interleaving-depth media type parameter as follows. When the value of the OPTIONAL sprop-interleaving-depth media type parameter is equal to 0 (explicitly or per default), the transmission order of NAL units MUST conform to the NAL unit decoding order. When the value of the OPTIONAL sprop-interleaving-depth media type parameter is greater than 0:

- the order of NAL units in an MTAP16 and an MTAP24 is not required to be the NAL unit decoding order, and
- the order of NAL units generated by de-packetizing STAP-Bs, MTAPs, and FUs in two consecutive packets is not required to be the NAL unit decoding order.

The RTP payload structures for a single NAL unit packet, an STAP-A, and an FU-A do not include DON. STAP-B and FU-B structures include DON, and the structure of MTAPs enables derivation of DON, as specified in Section 5.7.2.

> Informative note: When an FU-A occurs in interleaved mode, it always follows an FU-B, which sets its DON.
>
> Informative note: If a transmitter wants to encapsulate a single NAL unit per packet and transmit packets out of their decoding order, STAP-B packet type can be used.

In the single NAL unit packetization mode, the transmission order of NAL units, determined by the RTP sequence number, MUST be the same as their NAL unit decoding order. In the non-interleaved packetization

mode, the transmission order of NAL units in single NAL unit packets, STAP-As, and FU-As MUST be the same as their NAL unit decoding order. The NAL units within an STAP MUST appear in the NAL unit decoding order. Thus, the decoding order is first provided through the implicit order within an STAP and then provided through the RTP sequence number for the order between STAPs, FUs, and single NAL unit packets.

The signaling of the value of DON for NAL units carried in STAP-B, MTAP, and a series of fragmentation units starting with an FU-B is specified in Sections 5.7.1, 5.7.2, and 5.8, respectively. The DON value of the first NAL unit in transmission order MAY be set to any value. Values of DON are in the range of 0 to 65535, inclusive. After reaching the maximum value, the value of DON wraps around to 0.

The decoding order of two NAL units contained in any STAP-B, MTAP, or a series of fragmentation units starting with an FU-B is determined as follows. Let DON(i) be the decoding order number of the NAL unit having index i in the transmission order. Function don_diff(m,n) is specified as follows:

```
If DON(m) == DON(n), don_diff(m,n) = 0

If (DON(m) < DON(n) and DON(n) - DON(m) < 32768),
don_diff(m,n) = DON(n) - DON(m)

If (DON(m) > DON(n) and DON(m) - DON(n) >= 32768),
don_diff(m,n) = 65536 - DON(m) + DON(n)

If (DON(m) < DON(n) and DON(n) - DON(m) >= 32768),
don_diff(m,n) = - (DON(m) + 65536 - DON(n))

If (DON(m) > DON(n) and DON(m) - DON(n) < 32768),
don_diff(m,n) = - (DON(m) - DON(n))
```

A positive value of don_diff(m,n) indicates that the NAL unit having transmission order index n follows, in decoding order, the NAL unit having transmission order index m. When don_diff(m,n) is equal to 0, the NAL unit decoding order of the two NAL units can be in either order. A negative value of don_diff(m,n) indicates that the NAL unit having transmission order index n precedes, in decoding order, the NAL unit having transmission order index m.

Values of DON-related fields (DON, DONB, and DOND; see Section 5.7) MUST be such that the decoding order determined by the values of DON, as specified above, conforms to the NAL unit decoding order.

If the order of two NAL units in NAL unit decoding order is switched and the new order does not conform to the NAL unit decoding order, the NAL units MUST NOT have the same value of DON. If the order of two consecutive NAL units in the NAL unit stream is switched and the new order still conforms to the NAL unit decoding order, the NAL units MAY have the same value of DON. For example, when arbitrary slice order is allowed by the video coding profile in use, all the coded slice NAL units of a coded picture are allowed to have the same value of DON. Consequently, NAL units having the same value of DON can be decoded in any order, and two NAL units having a different value of DON should be passed to the decoder in the order specified above. When two consecutive NAL units in the NAL unit decoding order have a different value of DON, the value of DON for the second NAL unit in decoding order SHOULD be the value of DON for the first, incremented by one.

An example of the de-packetization process to recover the NAL unit decoding order is given in Section 7.

> Informative note: Receivers should not expect that the absolute difference of values of DON for two consecutive NAL units in the NAL unit decoding order will be equal to one, even in error-free transmission. An increment by one is not required, as at the time of associating values of DON to NAL units, it may not be known whether all NAL units are delivered to the receiver. For example, a gateway may not forward coded slice NAL units of non-reference pictures or SEI NAL units when there is a shortage of bitrate in the network to which the packets are forwarded. In another example, a live broadcast is interrupted by pre-encoded content, such as commercials, from time to time. The first intra picture of a pre-encoded clip is transmitted in advance to ensure that it is readily available in the receiver. When transmitting the first intra picture, the originator does not exactly know how many NAL units will be encoded before the first intra picture of the pre-encoded clip follows in decoding order. Thus, the values of DON for the NAL units of the first intra picture of the pre-encoded clip have to be estimated when they are transmitted, and gaps in values of DON may occur.

5.6. Single NAL Unit Packet

The single NAL unit packet defined here MUST contain only one NAL unit of the types defined in [1]. This means that neither an aggregation packet nor a fragmentation unit can be used within a single NAL unit packet. 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. The structure of the single NAL unit packet is shown in Figure 2.

Informative note: The first byte of a NAL unit co-serves as the RTP payload header.

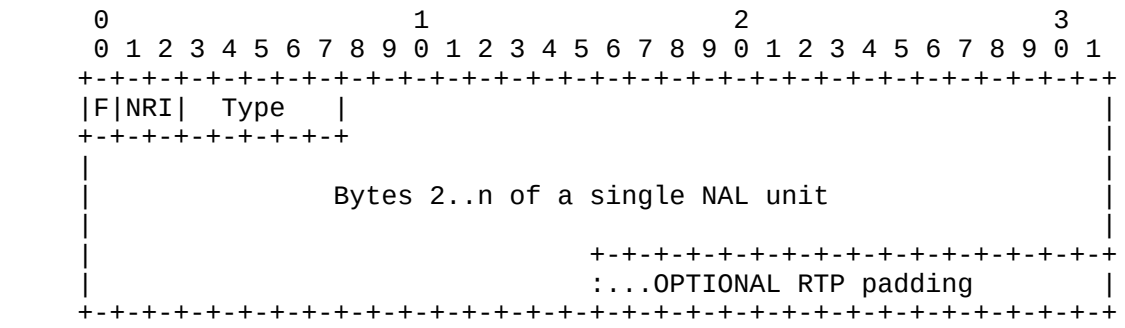


Figure 2. RTP payload format for single NAL unit packet

5.7. Aggregation Packets

Aggregation packets are the NAL unit aggregation scheme of this payload specification. The scheme is introduced to reflect the dramatically different MTU sizes of two key target networks: wireline IP networks (with an MTU size that is often limited by the Ethernet MTU size, roughly 1500 bytes) and IP-based or non-IP-based (e.g., ITU-T H.324/M) wireless communication systems with preferred transmission unit sizes of 254 bytes or less. To prevent media transcoding between the two worlds, and to avoid undesirable packetization overhead, a NAL unit aggregation scheme is introduced.

Two types of aggregation packets are defined by this specification:

- Single-time aggregation packet (STAP): aggregates NAL units with identical NALU-times. Two types of STAPs are defined, one without DON (STAP-A) and another including DON (STAP-B).

- Multi-time aggregation packet (MTAP): aggregates NAL units with potentially differing NALU-times. Two different MTAPs are defined, differing in the length of the NAL unit timestamp offset.

Each NAL unit to be carried in an aggregation packet is encapsulated in an aggregation unit. Please see below for the four different aggregation units and their characteristics.

The structure of the RTP payload format for aggregation packets is presented in Figure 3.

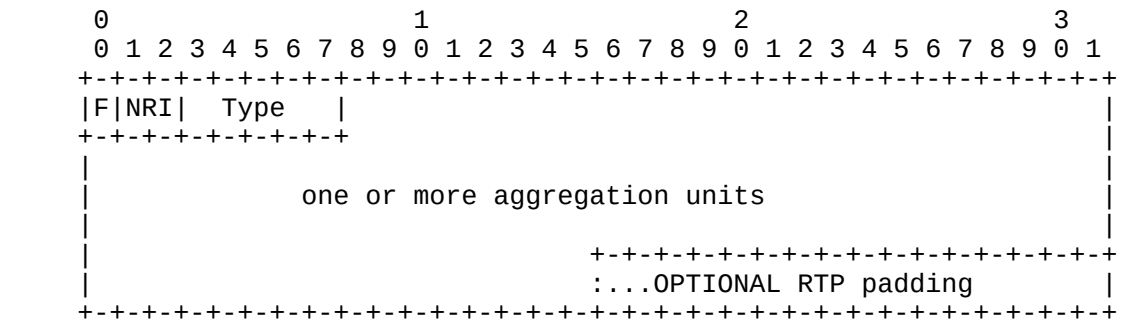


Figure 3. RTP payload format for aggregation packets

MTAPs and STAPs share the following packetization rules:

- The RTP timestamp MUST be set to the earliest of the NALU-times of all the NAL units to be aggregated.
- The type field of the NAL unit type octet MUST be set to the appropriate value, as indicated in Table 4.
- The F bit MUST be cleared if all F bits of the aggregated NAL units are zero; otherwise, it MUST be set.
- The value of NRI MUST be the maximum of all the NAL units carried in the aggregation packet.

Table 4. Type field for STAPs and MTAPs

Type	Packet	Timestamp offset field length (in bits)	DON-related fields (DON, DONB, DOND) present
24	STAP-A	0	no
25	STAP-B	0	yes
26	MTAP16	16	yes
27	MTAP24	24	yes

The marker bit in the RTP header is set to the value that the marker bit of the last NAL unit of the aggregated packet would have if it were transported in its own RTP packet.

The payload of an aggregation packet consists of one or more aggregation units. See Sections 5.7.1 and 5.7.2 for the four different types of aggregation units. An aggregation packet can carry as many aggregation units as necessary; however, the total amount of data in an aggregation packet obviously MUST fit into an IP packet, and the size SHOULD be chosen so that the resulting IP packet is smaller than the MTU size. An aggregation packet MUST NOT contain fragmentation units, as specified in Section 5.8. Aggregation packets MUST NOT be nested; that is, an aggregation packet MUST NOT contain another aggregation packet.

5.7.1. Single-Time Aggregation Packet (STAP)

A single-time aggregation packet (STAP) SHOULD be used whenever NAL units are aggregated that all share the same NALU-time. The payload of an STAP-A does not include DON and consists of at least one single-time aggregation unit, as presented in Figure 4. The payload of an STAP-B consists of a 16-bit unsigned decoding order number (DON) (in network byte order) followed by at least one single-time aggregation unit, as presented in Figure 5.

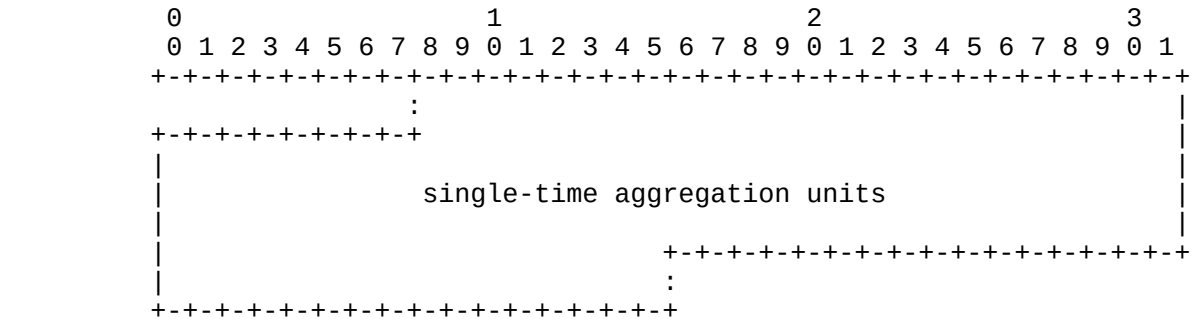


Figure 4. Payload format for STAP-A

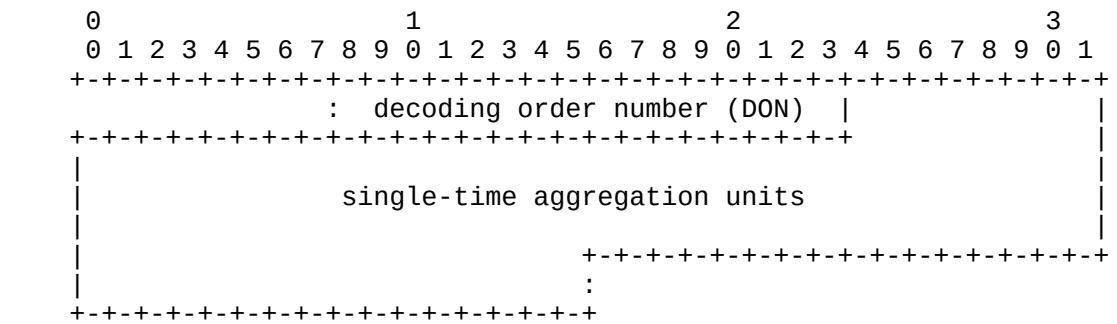


Figure 5. Payload format for STAP-B

The DON field specifies the value of DON for the first NAL unit in an STAP-B in transmission order. For each successive NAL unit in appearance order in an STAP-B, the value of DON is equal to (the value of DON of the previous NAL unit in the STAP-B + 1) % 65536, in which '%' stands for the modulo operation.

A single-time aggregation unit consists of 16-bit unsigned size information (in network byte order) that indicates the size of the following NAL unit in bytes (excluding these two octets, but including the NAL unit type octet of the NAL unit), followed by the NAL unit itself, including its NAL unit type byte. A single-time aggregation unit is byte aligned within the RTP payload, but it may not be aligned on a 32-bit word boundary. Figure 6 presents the structure of the single-time aggregation unit.

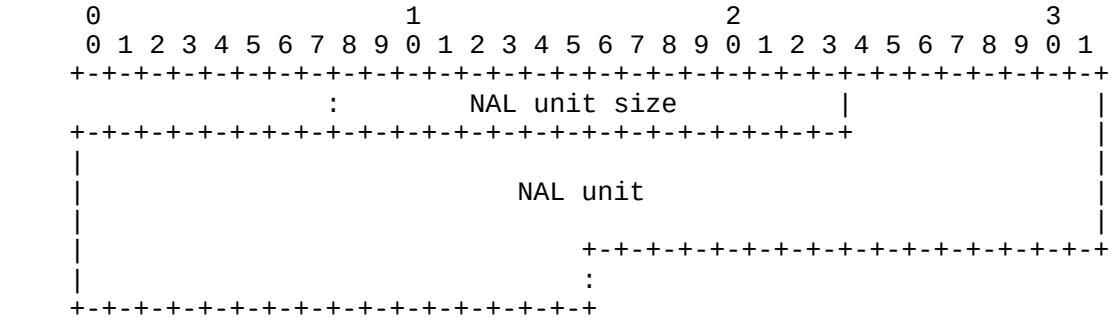


Figure 6. Structure for single-time aggregation unit

Figure 7 presents an example of an RTP packet that contains an STAP-A. The STAP contains two single-time aggregation units, labeled as 1 and 2 in the figure.

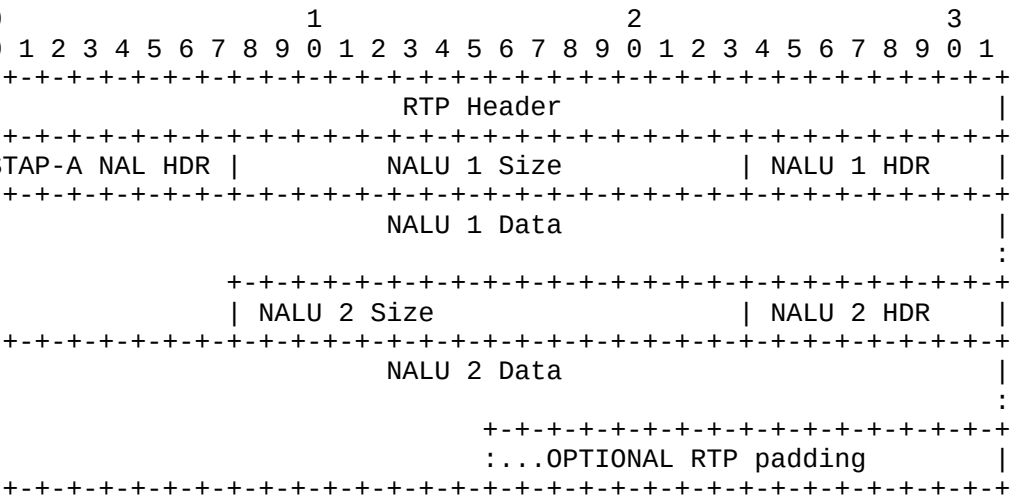


Figure 7. An example of an RTP packet including an STAP-A containing two single-time aggregation units

Figure 8 presents an example of an RTP packet that contains an STAP-B. The STAP contains two single-time aggregation units, labeled as 1 and 2 in the figure.

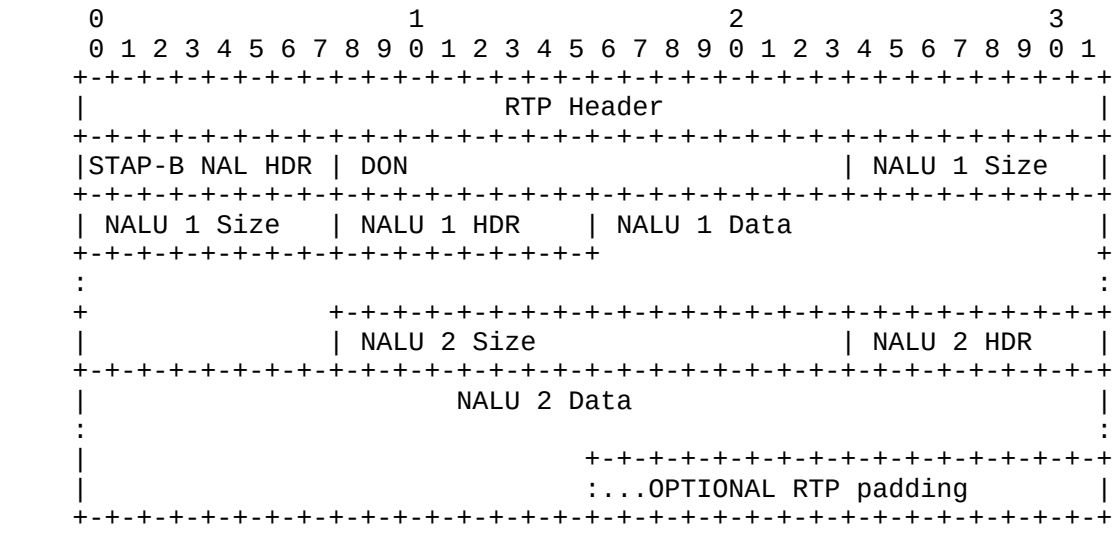


Figure 8. An example of an RTP packet including an STAP-B containing two single-time aggregation units

5.7.2. Multi-Time Aggregation Packets (MTAPs)

The NAL unit payload of MTAPs consists of a 16-bit unsigned decoding order number base (DONB) (in network byte order) and one or more multi-time aggregation units, as presented in Figure 9. DONB MUST contain the value of DON for the first NAL unit in the NAL unit decoding order among the NAL units of the MTAP.

> Informative note: The first NAL unit in the NAL unit decoding order is not necessarily the first NAL unit in the order in which the NAL units are encapsulated in an MTAP.

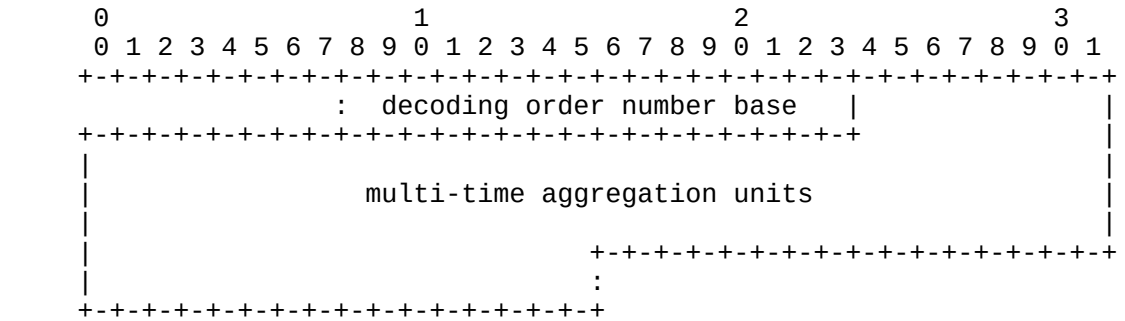


Figure 9. NAL unit payload format for MTAPs

Two different multi-time aggregation units are defined in this specification. Both of them consist of 16 bits of unsigned size information of the following NAL unit (in network byte order), an 8-bit unsigned decoding order number difference (DOND), and n bits (in network byte order) of timestamp offset (TS offset) for this NAL unit, whereby n can be 16 or 24. The choice between the different MTAP types (MTAP16 and MTAP24) is application dependent: the larger the timestamp offset is, the higher the flexibility of the MTAP, but the overhead is also higher.

The structure of the multi-time aggregation units for MTAP16 and MTAP24 are presented in Figures 10 and 11, respectively. The starting or ending position of an aggregation unit within a packet is not required to be on a 32-bit word boundary. The DON of the NAL unit contained in a multi-time aggregation unit is equal to (DONB + DOND) % 65536, in which % denotes the modulo operation. This memo does not specify how the NAL units within an MTAP are ordered, but, in most cases, NAL unit decoding order SHOULD be used.

The timestamp offset field MUST be set to a value equal to the value of the following formula: if the NALU-time is larger than or equal to the RTP timestamp of the packet, then the timestamp offset equals (the NALU-time of the NAL unit - the RTP timestamp of the packet). If the NALU-time is smaller than the RTP timestamp of the packet, then the timestamp offset is equal to the NALU-time + (2^32 - the RTP timestamp of the packet).

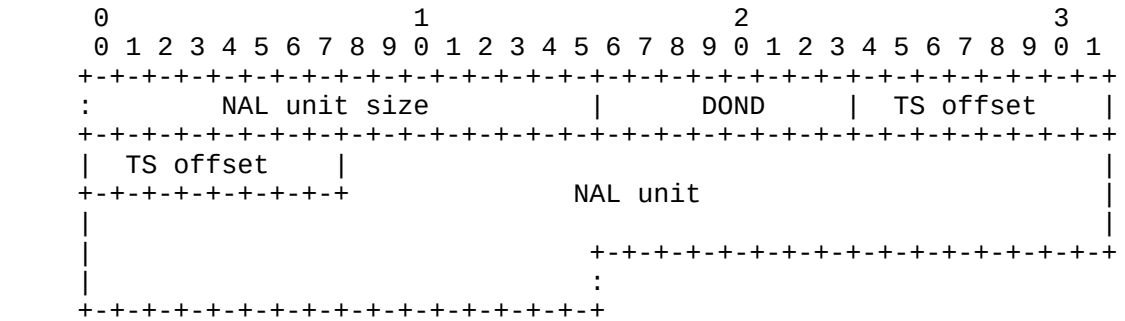


Figure 10. Multi-time aggregation unit for MTAP16

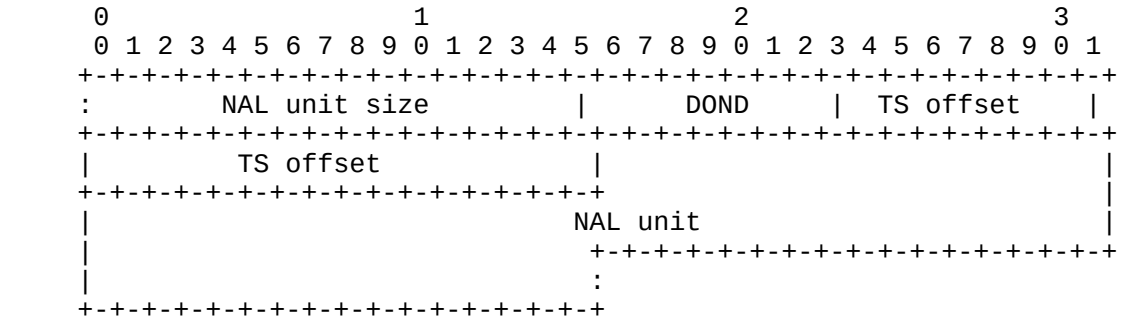


Figure 11. Multi-time aggregation unit for MTAP24

For the "earliest" multi-time aggregation unit in an MTAP, the timestamp offset MUST be zero. Hence, the RTP timestamp of the MTAP itself is identical to the earliest NALU-time.

Informative note: The "earliest" multi-time aggregation unit is the one that would have the smallest extended RTP timestamp among all the aggregation units of an MTAP if the NAL units contained in the aggregation units were encapsulated in single NAL unit packets. An extended timestamp is a timestamp that has more than 32 bits and is capable of counting the wraparound of the timestamp field, thus enabling one to determine the smallest value if the timestamp wraps. Such an "earliest" aggregation unit may not be the first one in the order in which the aggregation units are encapsulated in an MTAP. The "earliest" NAL unit need not be the same as the first NAL unit in the NAL unit decoding order either.

Figure 12 presents an example of an RTP packet that contains a multi-time aggregation packet of type MTAP16 that contains two multi-time aggregation units, labeled as 1 and 2 in the figure.

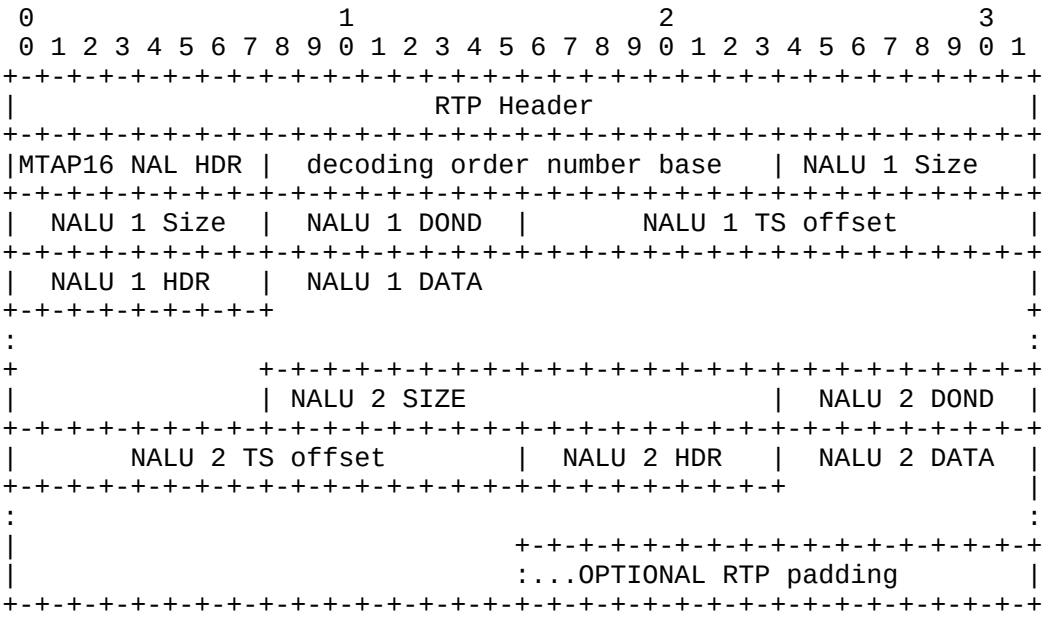


Figure 12. An RTP packet including a multi-time aggregation packet of type MTAP16 containing two multi-time aggregation units

Figure 13 presents an example of an RTP packet that contains a multi-time aggregation packet of type MTAP24 that contains two multi-time aggregation units, labeled as 1 and 2 in the figure.

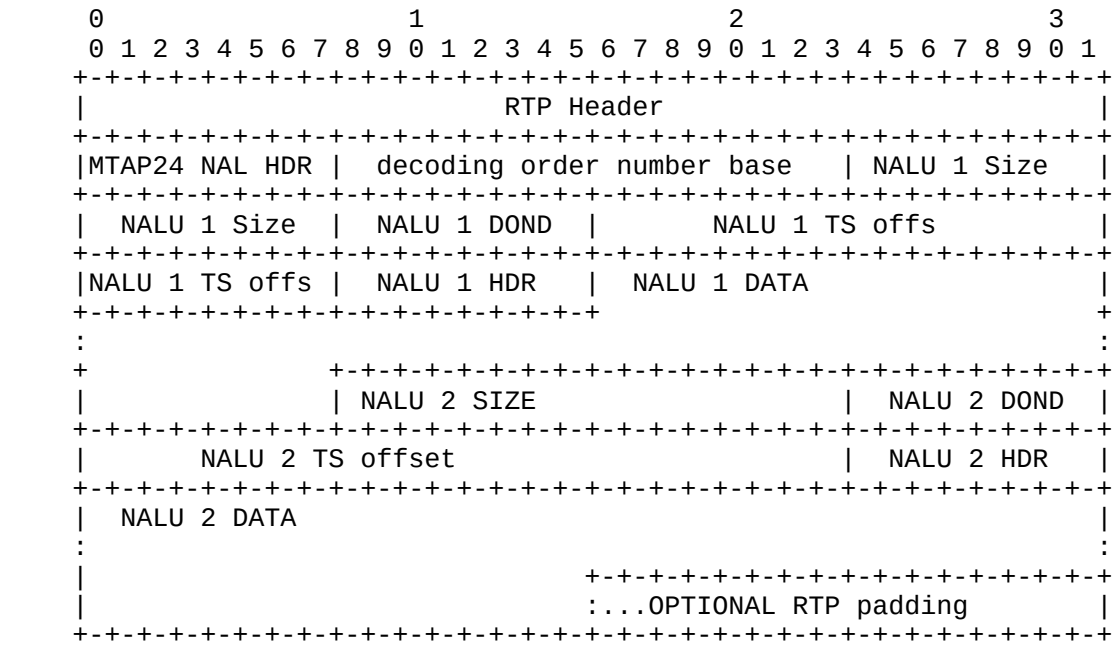


Figure 13. An RTP packet including a multi-time aggregation packet of type MTAP24 containing two multi-time aggregation units

5.8. Fragmentation Units (FUs)

This payload type allows fragmenting a NAL unit into several RTP packets. Doing so on the application layer instead of relying on lower-layer fragmentation (e.g., by IP) has the following advantages:

- The payload format is capable of transporting NAL units bigger than 64 kbytes over an IPv4 network that may be present in pre-recorded video, particularly in High-Definition formats (there is a limit of the number of slices per picture, which results in a limit of NAL units per picture, which may result in big NAL units).

- The fragmentation mechanism allows fragmenting a single NAL unit and applying generic forward error correction as described in Section 12.5.

Fragmentation is defined only for a single NAL unit and not for any aggregation packets. A fragment of a NAL unit consists of an integer number of consecutive octets of that NAL unit. Each octet of the NAL unit MUST be part of exactly one fragment 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 packet stream being sent between the first and last fragment). Similarly, a NAL unit MUST be reassembled in RTP sequence number order.

When a NAL unit is fragmented and conveyed within fragmentation units (FUs), it is referred to as a fragmented NAL unit. STAPs and MTAPs MUST NOT be fragmented. FUs MUST NOT be nested; that is, an FU MUST NOT contain another FU.

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

Figure 14 presents the RTP payload format for FU-As. An FU-A consists of a fragmentation unit indicator of one octet, a fragmentation unit header of one octet, and a fragmentation unit payload.

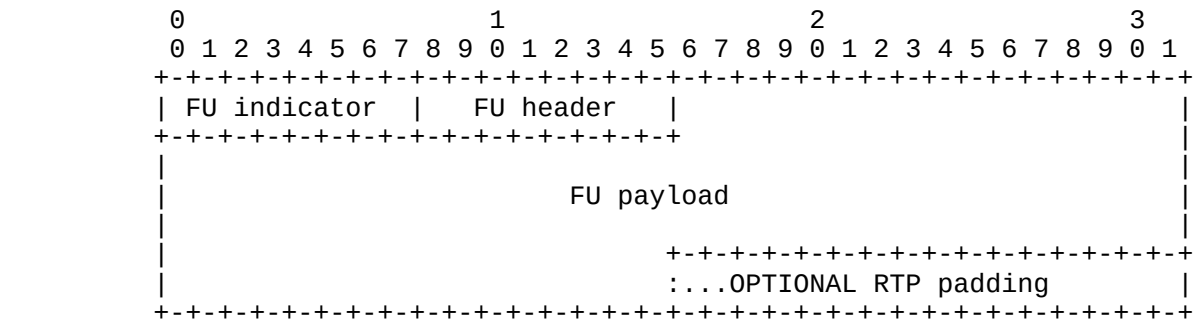


Figure 14. RTP payload format for FU-A

Figure 15 presents the RTP payload format for FU-Bs. An FU-B consists of a fragmentation unit indicator of one octet, a fragmentation unit header of one octet, a decoding order number (DON) (in network byte order), and a fragmentation unit payload. In other words, the structure of FU-B is the same as the structure of FU-A, except for the additional DON field.

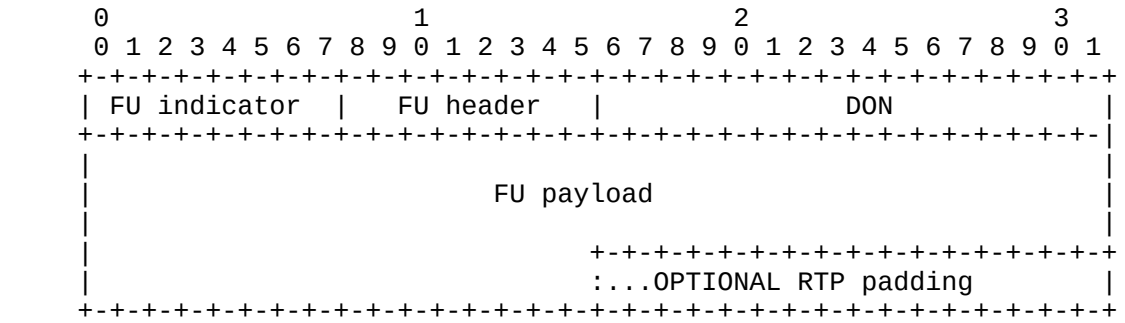
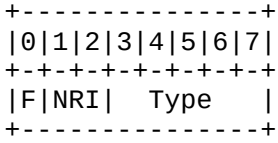


Figure 15. RTP payload format for FU-B

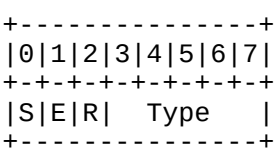
NAL unit type FU-B MUST be used in the interleaved packetization mode for the first fragmentation unit of a fragmented NAL unit. NAL unit type FU-B MUST NOT be used in any other case. In other words, in the interleaved packetization mode, each NALU that is fragmented has an FU-B as the first fragment, followed by one or more FU-A fragments.

The FU indicator octet has the following format:



Values equal to 28 and 29 in the type field of the FU indicator octet identify an FU-A and an FU-B, respectively. The use of the F bit is described in Section 5.3. The value of the NRI field MUST be set according to the value of the NRI field in the fragmented NAL unit.

The FU header has the following format:



S: 1 bit
When set to one, the Start bit indicates the start of a fragmented NAL unit. When the following FU payload is not the start of a fragmented NAL unit payload, the Start bit is set to zero.

E: 1 bit
When set to one, the End 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 following FU payload is not the last fragment of a fragmented NAL unit, the End bit is set to zero.

R: 1 bit
The Reserved bit MUST be equal to 0 and MUST be ignored by the receiver.

Type: 5 bits
The NAL unit payload type as defined in Table 7-1 of [1].

The value of DON in FU-Bs is selected as described in Section 5.5.

Informative note: The DON field in FU-Bs allows gateways to fragment NAL units to FU-Bs without organizing the incoming NAL units to the NAL unit decoding order.

A fragmented NAL unit MUST NOT be transmitted in one FU; that is, the Start bit and End bit MUST NOT both be set to one in the same FU header.

The FU payload consists of fragments of the payload of the fragmented NAL unit so that if the fragmentation unit payloads of consecutive FUs are sequentially concatenated, the payload of the fragmented NAL unit can be reconstructed. The NAL unit type octet of the fragmented NAL unit is not included as such in the fragmentation unit payload, but rather the information of the NAL unit type octet of the fragmented NAL unit is conveyed in the F and NRI fields of the FU indicator octet of the fragmentation unit and in the type field of the FU header. An FU payload MAY have any number of octets and MAY be empty.

Informative note: Empty FUs are allowed to reduce the latency of a certain class of senders in nearly lossless environments. These senders can be characterized in that they packetize NALU fragments before the NALU is completely generated and, hence, before the NALU size is known. If zero-length NALU fragments were not allowed, the sender would have to generate at least one bit of data of the following fragment before the current fragment could be sent. Due to the characteristics of H.264, where sometimes several macroblocks occupy zero bits, this is undesirable and can add delay. However, the (potential) use of zero-length NALU fragments should be carefully weighed against the increased risk of the loss of at least a part of the NALU because of the additional packets employed for its transmission.

If a fragmentation unit is lost, the receiver SHOULD discard all following fragmentation units in transmission order corresponding to the same fragmented NAL unit.

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

6. Packetization Rules

The packetization modes are introduced in Section 5.2. The packetization rules common to more than one of the packetization modes are specified in Section 6.1. The packetization rules for the single NAL unit mode, the non-interleaved mode, and the interleaved mode are specified in Sections 6.2, 6.3, and 6.4, respectively.

6.1. Common Packetization Rules

All senders MUST enforce the following packetization rules, regardless of the packetization mode in use:

- Coded slice NAL units or coded slice data partition NAL units belonging to the same coded picture (and thus sharing the same RTP timestamp value) MAY be sent in any order; however, for delay-critical systems, they SHOULD be sent in their original decoding order to minimize the delay. Note that the decoding order is the order of the NAL units in the bitstream.

- Parameter sets are handled in accordance with the rules and recommendations given in Section 8.4.

- MANEs MUST NOT duplicate any NAL unit except for sequence or picture parameter set NAL units, as neither this memo nor the H.264 specification provides means to identify duplicated NAL units. Sequence and picture parameter set NAL units MAY be duplicated to make their correct reception more probable, but any such duplication MUST NOT affect the contents of any active sequence or picture parameter set. Duplication SHOULD be performed on the application layer and not by duplicating RTP packets (with identical sequence numbers).

Senders using the non-interleaved mode and the interleaved mode MUST enforce the following packetization rule:

- In an RTP translator, MANEs MAY convert single NAL unit packets into one aggregation packet, convert an aggregation packet into several single NAL unit packets, or mix both concepts. The RTP translator SHOULD take into account at least the following parameters: path MTU size, unequal protection mechanisms (e.g., through packet-based FEC according to RFC 5109 [18], especially for sequence and picture parameter set NAL units and coded slice data partition A NAL units), bearable latency of the system, and buffering capabilities of the receiver.

  Informative note: An RTP translator is required to handle RTP Control Protocol (RTCP) as per RFC 3550.

6.2. Single NAL Unit Mode

This mode is in use when the value of the OPTIONAL packetization-mode media type parameter is equal to 0 or the packetization-mode is not present. All receivers MUST support this mode. It is primarily intended for low-delay applications that are compatible with systems using ITU-T Recommendation H.241 [3] (see Section 12.1). Only single NAL unit packets MAY be used in this mode. STAPs, MTAPs, and FUs MUST NOT be used. The transmission order of single NAL unit packets MUST comply with the NAL unit decoding order.

6.3. Non-Interleaved Mode

This mode is in use when the value of the OPTIONAL packetization-mode media type parameter is equal to 1. This mode SHOULD be supported. It is primarily intended for low-delay applications. Only single NAL unit packets, STAP-As, and FU-As MAY be used in this mode. STAP-Bs, MTAPs, and FU-Bs MUST NOT be used. The transmission order of NAL units MUST comply with the NAL unit decoding order.

6.4. Interleaved Mode

This mode is in use when the value of the OPTIONAL packetization-mode media type parameter is equal to 2. Some receivers MAY support this mode. STAP-Bs, MTAPs, FU-As, and FU-Bs MAY be used. STAP-As and single NAL unit packets MUST NOT be used. The transmission order of packets and NAL units is constrained as specified in Section 5.5.

7. De-Packetization Process

The de-packetization process is implementation dependent. Therefore, the following description should be seen as an example of a suitable implementation. Other schemes may also be used as long as the output for the same input is the same as the process described below. The same output means that the resulting NAL units and their order are identical. Optimizations relative to the described algorithms are likely possible. Section 7.1 presents the de-packetization process for the single NAL unit and non-interleaved packetization modes, whereas Section 7.2 describes the process for the interleaved mode. Section 7.3 includes additional de-packetization guidelines for intelligent receivers.

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.

7.1. Single NAL Unit and Non-Interleaved Mode

The receiver includes a receiver buffer to compensate for transmission delay jitter. The receiver stores incoming packets in reception order into the receiver buffer. Packets are de-packetized in RTP sequence number order. If a de-packetized packet is a single NAL unit packet, the NAL unit contained in the packet is passed directly to the decoder. If a de-packetized packet is an STAP-A, the NAL units contained in the packet are passed to the decoder in the order in which they are encapsulated in the packet. For all the FU-A packets containing fragments of a single NAL unit, the de-packetized fragments are concatenated in their sending order to recover the NAL unit, which is then passed to the decoder.

> Informative note: If the decoder supports arbitrary slice order, coded slices of a picture can be passed to the decoder in any order, regardless of their reception and transmission order.

7.2. Interleaved Mode

The general concept behind these de-packetization rules is to reorder NAL units from transmission order to the NAL unit decoding order.

The receiver includes a receiver buffer, which is used to compensate for transmission delay jitter and to reorder NAL units from transmission order to the NAL unit decoding order. In this section, the receiver operation is described under the assumption that there

is no transmission delay jitter. To differentiate the receiver buffer from a practical receiver buffer that is also used for compensation of transmission delay jitter, the receiver buffer is hereafter called the de-interleaving buffer in this section. Receivers SHOULD also prepare for transmission delay jitter, i.e., either reserve separate buffers for transmission delay jitter buffering and de-interleaving buffering or use a receiver buffer for both transmission delay jitter and de-interleaving. Moreover, receivers SHOULD take transmission delay jitter into account in the buffering operation, e.g., by additional initial buffering before starting of decoding and playback.

This section is organized as follows: Subsection 7.2.1 presents how to calculate the size of the de-interleaving buffer. Subsection 7.2.2 specifies the receiver process on how to organize received NAL units to the NAL unit decoding order.

7.2.1. Size of the De-Interleaving Buffer

In either Offer/Answer or declarative Session Description Protocol (SDP) usage, the sprop-deint-buf-req media type parameter signals the requirement for the de-interleaving buffer size. Therefore, it is RECOMMENDED to set the de-interleaving buffer size, in terms of number of bytes, equal to or greater than the value of the sprop-deint-buf-req media type parameter.

When the SDP Offer/Answer model or any other capability exchange procedure is used in session setup, the properties of the received stream SHOULD be such that the receiver capabilities are not exceeded. In the SDP Offer/Answer model, the receiver can indicate its capabilities to allocate a de-interleaving buffer with the deint-buf-cap media type parameter. See Section 8.1 for further information on the deint-buf-cap and sprop-deint-buf-req media type parameters and Section 8.2.2 for further information on their use in the SDP Offer/Answer model.

7.2.2. De-Interleaving Process

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

Regardless of the buffering state, the receiver stores incoming NAL units, in reception order, in the de-interleaving buffer as follows. NAL units of aggregation packets are stored in the de-interleaving buffer individually. The value of DON is calculated and stored for each NAL unit.

The receiver operation is described below with the help of the following functions and constants:

- Function AbsDON is specified in Section 8.1.
- Function don_diff is specified in Section 5.5.
- Constant N is the value of the OPTIONAL sprop-interleaving-depth media type parameter (see Section 8.1) incremented by 1.

Initial buffering lasts until one of the following conditions is fulfilled:

- There are N or more VCL NAL units in the de-interleaving buffer.
- If sprop-max-don-diff is present, don_diff(m,n) is greater than the value of sprop-max-don-diff, in which n corresponds to the NAL unit having the greatest value of AbsDON among the received NAL units and m corresponds to the NAL unit having the smallest value of AbsDON among the received NAL units.
- Initial buffering has lasted for the duration equal to or greater than the value of the OPTIONAL sprop-init-buf-time media type parameter.

The NAL units to be removed from the de-interleaving buffer are determined as follows:

- If the de-interleaving buffer contains at least N VCL NAL units, NAL units are removed from the de-interleaving buffer and passed to the decoder in the order specified below until the buffer contains N-1 VCL NAL units.
- If sprop-max-don-diff is present, all NAL units m for which don_diff(m,n) is greater than sprop-max-don-diff are removed from the de-interleaving buffer and passed to the decoder in the order specified below. Herein, n corresponds to the NAL unit having the greatest value of AbsDON among the NAL units in the de-interleaving buffer.

The order in which NAL units are passed to the decoder is specified as follows:

- Let PDON be a variable that is initialized to 0 at the beginning of the RTP session.

- For each NAL unit associated with a value of DON, a DON distance is calculated as follows. If the value of DON of the NAL unit is larger than the value of PDON, the DON distance is equal to DON - PDON. Otherwise, the DON distance is equal to 65535 - PDON + DON + 1.

- NAL units are delivered to the decoder in ascending order of DON distance. If several NAL units share the same value of DON distance, they can be passed to the decoder in any order.

- When a desired number of NAL units have been passed to the decoder, the value of PDON is set to the value of DON for the last NAL unit passed to the decoder.

7.3. Additional De-Packetization Guidelines

The following additional de-packetization rules may be used to implement an operational H.264 de-packetizer:

- Intelligent RTP receivers (e.g., in gateways) may identify lost coded slice data partitions A (DPAs). If a lost DPA is detected, after taking into account possible retransmission and FEC, a gateway may decide not to send the corresponding coded slice data partitions B and C, as their information is meaningless for H.264 decoders. In this way, a MANE can reduce network load by discarding useless packets without parsing a complex bitstream.

- Intelligent RTP receivers (e.g., in gateways) may identify lost FUs. If a lost FU is found, a gateway may decide not to send the following FUs of the same fragmented NAL unit, as their information is meaningless for H.264 decoders. In this way, a MANE can reduce network load by discarding useless packets without parsing a complex bitstream.

- Intelligent receivers having to discard packets or NALUs should first discard all packets/NALUs in which the value of the NRI field of the NAL unit type octet is equal to 0. This will minimize the impact on user experience and keep the reference pictures intact. If more packets have to be discarded, then

packets with a numerically lower NRI value should be discarded before packets with a numerically higher NRI value. However, discarding any packets with an NRI bigger than 0 very likely leads to decoder drift and SHOULD be avoided.

8. Payload Format Parameters

This section specifies the parameters that MAY be used to select optional features of the payload format and certain features of the bitstream. The parameters are specified here as part of the media subtype registration for the ITU-T H.264 | ISO/IEC 14496-10 codec. A mapping of the parameters into the Session Description Protocol (SDP) [6] is also provided for applications that use SDP. Equivalent parameters could be defined elsewhere for use with control protocols that do not use SDP.

Some parameters provide a receiver with the properties of the stream that will be sent. The names of all these parameters start with "sprop" for stream properties. Some of these "sprop" parameters are limited by other payload or codec configuration parameters. For example, the sprop-parameter-sets parameter is constrained by the profile-level-id parameter.

8.1. Media Type Registration

The media subtype for the ITU-T H.264 | ISO/IEC 14496-10 codec has been allocated from the IETF tree.

Media Type name: video

Media subtype name: H264

Required parameters: none

OPTIONAL parameters:

profile-level-id:
A base16 [7] (hexadecimal) representation of the following three bytes in the sequence parameter set NAL unit is specified in [1]: 1) profile_idc, 2) a byte herein referred to as profile-iop, composed of the values of constraint_set0_flag, constraint_set1_flag, constraint_set2_flag, constraint_set3_flag, constraint_set4_flag, constraint_set5_flag, and reserved_zero_2bits in bit-significance order, starting from the most-significant bit, and 3) level_idc. Note that reserved_zero_2bits is required to be equal to 0 in [1], but other values for it may be specified in the future by ITU-T or ISO/IEC.

The profile-level-id parameter indicates the default sub-profile (i.e., the subset of coding tools that may have been used to generate the stream or that the receiver supports) and the default level of the stream or the receiver supports.

The default sub-profile is indicated collectively by the profile_idc byte and some fields in the profile-iop byte. Depending on the values of the fields in the profile-iop byte, the default sub-profile may be the set of coding tools supported by one profile, or a common subset of coding tools of multiple profiles, as specified in Section 7.4.2.1.1 of [1]. The default level is indicated by the level_idc byte, and, when profile_idc is equal to 66, 77, or 88 (the Baseline, Main, or Extended profile) and level_idc is equal to 11, additionally by bit 4 (constraint_set3_flag) of the profile-iop byte. When profile_idc is equal to 66, 77, or 88 (the Baseline, Main, or Extended profile), level_idc is equal to 11, and bit 4 (constraint_set3_flag) of the profile-iop byte is equal to 1, the default level is Level 1b.

Table 5 lists all profiles defined in Annex A of [1] and, for each of the profiles, the possible combinations of profile_idc and profile-iop that represent the same sub-profile.

Table 5. Combinations of profile_idc and profile-iop representing the same sub-profile corresponding to the full set of coding tools supported by one profile. In the following, x may be either 0 or 1, while the profile names are indicated as follows. CB: Constrained Baseline profile, B: Baseline profile, M: Main profile, E: Extended profile, H: High profile, H10: High 10 profile, H42: High 4:2:2 profile, H44: High 4:4:4 Predictive profile, H10I: High 10 Intra profile, H42I: High 4:2:2 Intra profile, H44I: High 4:4:4 Intra profile, and C44I: CAVLC 4:4:4 Intra profile.

Profile	profile_idc (hexadecimal)	profile-iop (binary)
CB	42 (B)	x1xx0000
same as:	4D (M)	1xxx0000
same as:	58 (E)	11xx0000
B	42 (B)	x0xx0000
same as:	58 (E)	10xx0000
M	4D (M)	0x0x0000
E	58	00xx0000
H	64	00000000
H10	6E	00000000
H42	7A	00000000
H44	F4	00000000
H10I	6E	00010000
H42I	7A	00010000
H44I	F4	00010000
C44I	2C	00010000

For example, in the table above, profile_idc equal to 58 (Extended) with profile-iop equal to 11xx0000 indicates the same sub-profile corresponding to profile_idc equal to 42 (Baseline) with profile-iop equal to x1xx0000. Note that other combinations of profile_idc and profile-iop (not listed in Table 5) may represent a sub-profile equivalent to the common subset of coding tools for more than one profile. Note also that a decoder conforming to a certain profile may be able to decode bitstreams conforming to other profiles.

If the profile-level-id parameter is used to indicate properties of a NAL unit stream, it indicates that, to decode the stream, the minimum subset of coding tools a decoder has to support is the default sub-profile, and the lowest level the decoder has to support is the default level.

If the profile-level-id parameter is used for capability exchange or session setup, it indicates the subset of coding tools, which is equal to the default sub-profile, that the codec supports for both receiving and sending. If max-recv-level is not present, the default level from profile-level-id indicates the highest level the codec wishes to support. If max-recv-level is present, it indicates the highest level the codec supports for receiving. For either receiving or sending, all levels that are lower than the highest level supported MUST also be supported.

Informative note: Capability exchange and session setup procedures should provide means to list the capabilities for each supported sub-profile separately. For example, the one-of-N codec selection procedure of the SDP Offer/Answer model can be used (Section 10.2 of [8]). The one-of-N codec selection procedure may also be used to provide different combinations of profile_idc and profile-iop that represent the same sub-profile. When there are many different combinations of profile_idc and profile-iop that represent the same sub-profile, using the one-of-N codec selection procedure may result in a fairly large SDP message. Therefore, a receiver should understand the different equivalent combinations of profile_idc and profile-iop that represent the same sub-profile and be ready to accept an offer using any of the equivalent combinations.

If no profile-level-id is present, the Baseline profile, without additional constraints at Level 1, MUST be inferred.

max-recv-level:
This parameter MAY be used to indicate the highest level a receiver supports when the highest level is higher than the default level (the level indicated by profile-level-id). The value of max-recv-level is a base16 (hexadecimal) representation of the two bytes after the syntax element profile_idc in the sequence parameter set NAL unit specified in [1]: profile-iop (as defined above) and level_idc. If the level_idc byte of max-recv-level is equal to 11 and bit 4 of the profile-iop byte of max-recv-level is equal to 1 or if the level_idc byte of max-recv-level is equal to 9 and bit 4 of the profile-iop byte of max-recv-level is equal to 0, the highest level the receiver supports is Level 1b. Otherwise, the highest level the receiver supports is equal to the level_idc byte of max-recv-level divided by 10.

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

max-mbps, max-smbps, max-fs, max-cpb, max-dpb, and max-br:
These parameters MAY be used to signal the capabilities of a receiver implementation. These parameters MUST NOT be used for any other purpose. The highest level conveyed in the value of the profile-level-id parameter or the max-recv-level parameter MUST be such that the receiver is fully capable of supporting. max-mbps, max-smbps, max-fs, max-cpb, max-dpb, and max-br MAY be used to indicate capabilities of the receiver that extend the required capabilities of the signaled highest level, as specified below.

When more than one parameter from the set (max-mbps, max-smbps, max-fs, max-cpb, max-dpb, max-br) is present, the receiver MUST support all signaled capabilities simultaneously. For example, if both max-mbps and max-br are present, the signaled highest level with the extension of both the frame rate and bitrate is supported. That is, the receiver is able to decode NAL unit streams in which the macroblock processing rate is up to max-mbps (inclusive), the bitrate is up to max-br (inclusive), the coded picture buffer size is derived as specified in the semantics of the max-br parameter below, and the other properties comply with the highest level specified in the value of the profile-level-id parameter or the max-recv-level parameter.

If a receiver can support all the properties of Level A, the highest level specified in the value of the profile-level-id parameter or the max-recv-level parameter MUST be Level A (i.e., MUST NOT be lower than Level A). In other words, a receiver MUST NOT signal values of max-mbps, max-fs, max-cpb, max-dpb, and max-br that taken together meet the requirements of a higher level compared to the highest level specified in the value of the profile-level-id parameter or the max-recv-level parameter.

Informative note: When the OPTIONAL media type parameters are used to signal the properties of a NAL unit stream, max-mbps, max-smbps, max-fs, max-cpb, max-dpb, and max-br are not present, and the value of profile-level-id must always be such that the NAL unit stream complies fully with the specified profile and level.

max-mbps: The value of max-mbps is an integer indicating the maximum macroblock processing rate in units of macroblocks per second. The max-mbps parameter signals that the receiver is capable of decoding video at a higher rate than is required by the signaled highest level conveyed in the value of the profile-level-id parameter or the max-recv-level parameter.

When max-mbps is signaled, the receiver MUST be able to decode NAL unit streams that conform to the signaled highest level, with the exception that the MaxMBPS value in Table A-1 of [1] for the signaled highest level is replaced with the value of max-mbps. The value of max-mbps MUST be greater than or equal to the value of MaxMBPS given in Table A-1 of [1] for the highest level. Senders MAY use this knowledge to send pictures of a given size at a higher picture rate than is indicated in the signaled highest level.

max-smbps: The value of max-smbps is an integer indicating the maximum static macroblock processing rate in units of static macroblocks per second, under the hypothetical assumption that all macroblocks are static macroblocks. When max-smbps is signaled, the MaxMBPS value in Table A-1 of [1] should be replaced with the result of the following computation:

- If the parameter max-mbps is signaled, set a variable MaxMacroblocksPerSecond to the value of max-mbps. Otherwise, set MaxMacroblocksPerSecond equal to the value of MaxMBPS in Table A-1 [1] for the signaled highest level conveyed in the value of the profile-level-id parameter or the max-recv-level parameter.

- Set a variable P_non-static to the proportion of non-static macroblocks in picture n.

- Set a variable P_static to the proportion of static macroblocks in picture n.

- The value of MaxMBPS in Table A-1 of [1] should be considered by the encoder to be equal to:

  MaxMacroblocksPerSecond * max-smbps / (P_non-static * max-smbps + P_static * MaxMacroblocksPerSecond)

The encoder should recompute this value for each picture. The value of max-smbps MUST be greater than or equal to the value of MaxMBPS given explicitly as the value of the max-mbps parameter or implicitly in Table A-1 of [1] for the signaled highest level. Senders MAY use this knowledge to send pictures of a given size at a higher picture rate than is indicated in the signaled highest level.

max-fs: The value of max-fs is an integer indicating the maximum frame size in units of macroblocks. The max-fs parameter signals that the receiver is capable of decoding larger picture sizes than are required by the signaled highest level conveyed

in the value of the profile-level-id parameter or the max-recv-level parameter. When max-fs is signaled, the receiver MUST be able to decode NAL unit streams that conform to the signaled highest level, with the exception that the MaxFS value in Table A-1 of [1] for the signaled highest level is replaced with the value of max-fs. The value of max-fs MUST be greater than or equal to the value of MaxFS given in Table A-1 of [1] for the highest level. Senders MAY use this knowledge to send larger pictures at a proportionally lower frame rate than is indicated in the signaled highest level.

max-cpb: The value of max-cpb is an integer indicating the maximum coded picture buffer size in units of 1000 bits for the VCL HRD parameters and in units of 1200 bits for the NAL HRD parameters. Note that this parameter does not use units of cpbBrVclFactor and cpbBrNALFactor (see Table A-1 of [1]). The max-cpb parameter signals that the receiver has more memory than the minimum amount of coded picture buffer memory required by the signaled highest level conveyed in the value of the profile-level-id parameter or the max-recv-level parameter. When max-cpb is signaled, the receiver MUST be able to decode NAL unit streams that conform to the signaled highest level, with the exception that the MaxCPB value in Table A-1 of [1] for the signaled highest level is replaced with the value of max-cpb (after taking cpbBrVclFactor and cpbBrNALFactor into consideration when needed). The value of max-cpb (after taking cpbBrVclFactor and cpbBrNALFactor into consideration when needed) MUST be greater than or equal to the value of MaxCPB given in Table A-1 of [1] for the highest level. Senders MAY use this knowledge to construct coded video streams with greater variation of bitrate than can be achieved with the MaxCPB value in Table A-1 of [1].

Informative note: The coded picture buffer is used in the hypothetical reference decoder (Annex C of H.264). The use of the hypothetical reference decoder is recommended in H.264 encoders to verify that the produced bitstream conforms to the standard and to control the output bitrate. Thus, the coded picture buffer is conceptually independent of any other potential buffers in the receiver, including de-interleaving and de-jitter buffers. The coded picture buffer need not be implemented in decoders as specified in Annex C of H.264, but rather standard-compliant decoders can have any buffering arrangements provided that they can decode standard-compliant bitstreams. Thus, in practice, the input buffer for a video decoder can be integrated with de-interleaving and de-jitter buffers of the receiver.

max-dpb: The value of max-dpb is an integer indicating the maximum decoded picture buffer size in units of 8/3 macroblocks. The max-dpb parameter signals that the receiver has more memory than the minimum amount of decoded picture buffer memory required by the signaled highest level conveyed in the value of the profile-level-id parameter or the max-recv-level parameter. When max-dpb is signaled, the receiver MUST be able to decode NAL unit streams that conform to the signaled highest level, with the exception that the MaxDpbMbs value in Table A-1 of [1] for the signaled highest level is replaced with the value of max-dpb * 3 / 8. Consequently, a receiver that signals max-dpb MUST be capable of storing the following number of decoded frames, complementary field pairs, and non-paired fields in its decoded picture buffer:

Min(max-dpb * 3 / 8 / (PicWidthInMbs * FrameHeightInMbs), 16)

Wherein PicWidthInMbs and FrameHeightInMbs are defined in [1].

The value of max-dpb MUST be greater than or equal to the value of MaxDpbMbs * 3 / 8, wherein the value of MaxDpbMbs is given in Table A-1 of [1] for the highest level. Senders MAY use this knowledge to construct coded video streams with improved compression.

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

Informative note: In RFC 3984, which this document obsoletes, the unit of this parameter was 1024 bytes. The unit has been changed to 8/3 macroblocks in this document. The reason for this change was due to the changes from the 2003 version of the H.264 specification referenced by RFC 3984 to the 2010 version of the H.264 specification referenced by this document, particularly the changes to Table A-1 in the H.264 specification due to addition of color formats and bit depths not supported earlier. The changed semantics of this parameter keeps backward compatibility to RFC 3984 and supports all profiles defined in the 2010 version of the H.264 specification.

max-br: The value of max-br is an integer indicating the maximum video bitrate in units of 1000 bits per second for the VCL HRD parameters and in units of 1200 bits per second for the NAL HRD parameters. Note that this parameter does not use units of cpbBrVclFactor and cpbBrNALFactor (see Table A-1 of [1]).

The max-br parameter signals that the video decoder of the receiver is capable of decoding video at a higher bitrate than is required by the signaled highest level conveyed in the value of the profile-level-id parameter or the max-recv-level parameter.

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

- The value of max-br (after taking cpbBrVclFactor and cpbBrNALFactor into consideration when needed) replaces the MaxBR value in Table A-1 of [1] for the highest level.

- When the max-cpb parameter is not present, the result of the following formula replaces the value of MaxCPB in Table A-1 of [1]: (MaxCPB of the signaled level) * max-br / (MaxBR of the signaled highest level).

For example, if a receiver signals capability for Main profile Level 1.2 with max-br equal to 1550, this indicates a maximum video bitrate of 1550 kbits/sec for VCL HRD parameters, a maximum video bitrate of 1860 kbits/sec for NAL HRD parameters, and a CPB size of 4036458 bits (1550000 / 384000 * 1000 * 1000).

The value of max-br (after taking cpbBrVclFactor and cpbBrNALFactor into consideration when needed) MUST be greater than or equal to the value MaxBR given in Table A-1 of [1] for the signaled highest level.

Senders MAY use this knowledge to send higher bitrate video as allowed in the level definition of Annex A of H.264 to achieve improved video quality.

Informative note: This parameter was added primarily to complement a similar codepoint in the ITU-T Recommendation H.245, so as to facilitate signaling gateway designs. The assumption that the network is capable of handling such bitrates at any given time cannot be made from the value of

this parameter. In particular, no conclusion can be drawn that the signaled bitrate is possible under congestion control constraints.

redundant-pic-cap:
This parameter signals the capabilities of a receiver implementation. When equal to 0, the parameter indicates that the receiver makes no attempt to use redundant coded pictures to correct incorrectly decoded primary coded pictures. When equal to 0, the receiver is not capable of using redundant slices; therefore, a sender SHOULD avoid sending redundant slices to save bandwidth. When equal to 1, the receiver is capable of decoding any such redundant slice that covers a corrupted area in a primary decoded picture (at least partly), and therefore a sender MAY send redundant slices. When the parameter is not present, a value of 0 MUST be used for redundant-pic-cap. When present, the value of redundant-pic-cap MUST be either 0 or 1.

When the profile-level-id parameter is present in the same signaling as the redundant-pic-cap parameter and the profile indicated in profile-level-id is such that it disallows the use of redundant coded pictures (e.g., Main profile), the value of redundant-pic-cap MUST be equal to 0. When a receiver indicates redundant-pic-cap equal to 0, the received stream SHOULD NOT contain redundant coded pictures.

Informative note: Even if redundant-pic-cap is equal to 0, the decoder is able to ignore redundant codec pictures provided that the decoder supports a profile (Baseline, Extended) in which redundant coded pictures are allowed.

Informative note: Even if redundant-pic-cap is equal to 1, the receiver may also choose other error concealment strategies to replace or complement decoding of redundant slices.

sprop-parameter-sets:
This parameter MAY be used to convey any sequence and picture parameter set NAL units (herein referred to as the initial parameter set NAL units) that can be placed in the NAL unit stream to precede any other NAL units in decoding order. The parameter MUST NOT be used to indicate codec capability in any capability exchange procedure. The value of the parameter is a comma-separated (',') list of base64 [7] representations of parameter set NAL units as specified in Sections 7.3.2.1 and

7.3.2.2 of [1]. Note that the number of bytes in a parameter set NAL unit is typically less than 10, but a picture parameter set NAL unit can contain several hundred bytes.

Informative note: When several payload types are offered in the SDP Offer/Answer model, each with its own sprop-parameter-sets parameter, the receiver cannot assume that those parameter sets do not use conflicting storage locations (i.e., identical values of parameter set identifiers). Therefore, a receiver should buffer all sprop-parameter-sets and make them available to the decoder instance that decodes a certain payload type.

The sprop-parameter-sets parameter MUST only contain parameter sets that are conforming to the profile-level-id, i.e., the subset of coding tools indicated by any of the parameter sets MUST be equal to the default sub-profile, and the level indicated by any of the parameter sets MUST be equal to the default level.

sprop-level-parameter-sets:
This parameter MAY be used to convey any sequence and picture parameter set NAL units (herein referred to as the initial parameter set NAL units) that can be placed in the NAL unit stream to precede any other NAL units in decoding order and that are associated with one or more levels different than the default level. The parameter MUST NOT be used to indicate codec capability in any capability exchange procedure.

The sprop-level-parameter-sets parameter contains parameter sets for one or more levels that are different than the default level. All parameter sets associated with one level are clustered and prefixed with a three-byte field that has the same syntax as profile-level-id. This enables the receiver to install the parameter sets for one level and discard the rest. The three-byte field is named PLId, and all parameter sets associated with one level are named PSL, which has the same syntax as sprop-parameter-sets. Parameter sets for each level are represented in the form of PLId:PSL, i.e., PLId followed by a colon (':') and the base64 [7] representation of the initial parameter set NAL units for the level. Each pair of PLId:PSLs is also separated by a colon. Note that a PSL can contain multiple parameter sets for that level, separated with commas (',').

The subset of coding tools indicated by each PLId field MUST be equal to the default sub-profile, and the level indicated by each PLId field MUST be different than the default level. All

sequence parameter sets contained in each PSL MUST have the three bytes from profile_idc to level_idc, inclusive, equal to the preceding PLId.

Informative note: This parameter allows for efficient level downgrade or upgrade in SDP Offer/Answer and out-of-band transport of parameter sets simultaneously.

use-level-src-parameter-sets:
This parameter MAY be used to indicate a receiver capability. The value MAY be equal to either 0 or 1. When the parameter is not present, the value MUST be inferred to be equal to 0. The value 0 indicates that the receiver does not understand the sprop-level-parameter-sets parameter, does not understand the "fmtp" source attribute as specified in Section 6.3 of [9], will ignore sprop-level-parameter-sets when present, and will ignore sprop-parameter-sets when conveyed using the "fmtp" source attribute. The value 1 indicates that the receiver understands the sprop-level-parameter-sets parameter, understands the "fmtp" source attribute as specified in Section 6.3 of [9], and is capable of using parameter sets contained in the sprop-level-parameter-sets or contained in the sprop-parameter-sets that is conveyed using the "fmtp" source attribute.

Informative note: An RFC 3984 receiver does not understand sprop-level-parameter-sets, use-level-src-parameter-sets, or the "fmtp" source attribute as specified in Section 6.3 of [9]. Therefore, during SDP Offer/Answer, an RFC 3984 receiver as the answerer will simply ignore sprop-level-parameter-sets when present in an offer and sprop-parameter-sets conveyed using the "fmtp" source attribute, as specified in Section 6.3 of [9]. Assume that the offered payload type was accepted at a level lower than the default level. If the offered payload type included sprop-level-parameter-sets or included sprop-parameter-sets conveyed using the "fmtp" source attribute and if the offerer sees that the answerer has not included use-level-src-parameter-sets equal to 1 in the answer, the offerer knows that in-band transport of parameter sets is needed.

in-band-parameter-sets:
This parameter MAY be used to indicate a receiver capability. The value MAY be equal to either 0 or 1. The value 1 indicates that the receiver discards out-of-band parameter sets in sprop-parameter-sets and sprop-level-parameter-sets; therefore, the sender MUST transmit all parameter sets in-band. The value 0 indicates that the receiver utilizes out-of-band parameter sets

included in sprop-parameter-sets and/or sprop-level-parameter-sets. However, in this case, the sender MAY still choose to send parameter sets in-band. When in-band-parameter-sets is equal to 1, use-level-src-parameter-sets MUST NOT be present or MUST be equal to 0. When the parameter is not present, this receiver capability is not specified, and therefore the sender MAY send out-of-band parameter sets only, it MAY send in-band-parameter-sets only, or it MAY send both.

level-asymmetry-allowed:
This parameter MAY be used in SDP Offer/Answer to indicate whether level asymmetry, i.e., sending media encoded at a different level in the offerer-to-answerer direction than the level in the answerer-to-offerer direction, is allowed. The value MAY be equal to either 0 or 1. When the parameter is not present, the value MUST be inferred to be equal to 0. The value 1 in both the offer and the answer indicates that level asymmetry is allowed. The value of 0 in either the offer or the answer indicates that level asymmetry is not allowed.

If level-asymmetry-allowed is equal to 0 (or not present) in either the offer or the answer, level asymmetry is not allowed. In this case, the level to use in the direction from the offerer to the answerer MUST be the same as the level to use in the opposite direction.

packetization-mode:
This parameter signals the properties of an RTP payload type or the capabilities of a receiver implementation. Only a single configuration point can be indicated; thus, when capabilities to support more than one packetization-mode are declared, multiple configuration points (RTP payload types) must be used.

When the value of packetization-mode is equal to 0 or packetization-mode is not present, the single NAL mode MUST be used. This mode is in use in standards using ITU-T Recommendation H.241 [3] (see Section 12.1). When the value of packetization-mode is equal to 1, the non-interleaved mode MUST be used. When the value of packetization-mode is equal to 2, the interleaved mode MUST be used. The value of packetization-mode MUST be an integer in the range of 0 to 2, inclusive.

sprop-interleaving-depth:
This parameter MUST NOT be present when packetization-mode is not present or the value of packetization-mode is equal to 0 or 1. This parameter MUST be present when the value of packetization-mode is equal to 2.

This parameter signals the properties of an RTP packet stream. It specifies the maximum number of VCL NAL units that precede any VCL NAL unit in the RTP packet stream in transmission order and that follow the VCL NAL unit in decoding order. Consequently, it is guaranteed that receivers can reconstruct NAL unit decoding order when the buffer size for NAL unit decoding order recovery is at least the value of sprop-interleaving-depth + 1 in terms of VCL NAL units.

The value of sprop-interleaving-depth MUST be an integer in the range of 0 to 32767, inclusive.

sprop-deint-buf-req:
This parameter MUST NOT be present when packetization-mode is not present or the value of packetization-mode is equal to 0 or 1. It MUST be present when the value of packetization-mode is equal to 2.

sprop-deint-buf-req signals the required size of the de-interleaving buffer for the RTP packet stream. The value of the parameter MUST be greater than or equal to the maximum buffer occupancy (in units of bytes) required in such a de-interleaving buffer that is specified in Section 7.2. It is guaranteed that receivers can perform the de-interleaving of interleaved NAL units into NAL unit decoding order, when the de-interleaving buffer size is at least the value of sprop-deint-buf-req in terms of bytes.

The value of sprop-deint-buf-req MUST be an integer in the range of 0 to 4294967295, inclusive.

Informative note: sprop-deint-buf-req indicates the required size of the de-interleaving buffer only. When network jitter can occur, an appropriately sized jitter buffer has to be provisioned for as well.

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

If the parameter is not present, then a value of 0 MUST be used for deint-buf-cap. The value of deint-buf-cap MUST be an integer in the range of 0 to 4294967295, inclusive.

Informative note: deint-buf-cap indicates the maximum possible size of the de-interleaving buffer of the receiver only. When network jitter can occur, an appropriately sized jitter buffer has to be provisioned for as well.

sprop-init-buf-time:
This parameter MAY be used to signal the properties of an RTP packet stream. The parameter MUST NOT be present if the value of packetization-mode is equal to 0 or 1.

The parameter signals the initial buffering time that a receiver MUST wait before starting decoding to recover the NAL unit decoding order from the transmission order. The parameter is the maximum value of (decoding time of the NAL unit - transmission time of a NAL unit), assuming reliable and instantaneous transmission, the same timeline for transmission and decoding, and commencement of decoding when the first packet arrives.

An example of specifying the value of sprop-init-buf-time follows. A NAL unit stream is sent in the following interleaved order, in which the value corresponds to the decoding time and the transmission order is from left to right:

0 2 1 3 5 4 6 8 7 ...

Assuming a steady transmission rate of NAL units, the transmission times are:

0 1 2 3 4 5 6 7 8 ...

Subtracting the decoding time from the transmission time column-wise results in the following series:

0 -1 1 0 -1 1 0 -1 1 ...

Thus, in terms of intervals of NAL unit transmission times, the value of sprop-init-buf-time in this example is 1. The parameter is coded as a non-negative base10 integer representation in clock ticks of a 90-kHz clock. If the parameter is not present, then no initial buffering time value is defined. Otherwise, the value of sprop-init-buf-time MUST be an integer in the range of 0 to 4294967295, inclusive.

In addition to the signaled sprop-init-buf-time, receivers SHOULD take into account the transmission delay jitter buffering, including buffering for the delay jitter caused by mixers, translators, gateways, proxies, traffic-shapers, and other network elements.

sprop-max-don-diff:
This parameter MAY be used to signal the properties of an RTP packet stream. It MUST NOT be used to signal transmitter, receiver, or codec capabilities. The parameter MUST NOT be present if the value of packetization-mode is equal to 0 or 1. sprop-max-don-diff is an integer in the range of 0 to 32767, inclusive. If sprop-max-don-diff is not present, the value of the parameter is unspecified. sprop-max-don-diff is calculated as follows:

```
sprop-max-don-diff = max{AbsDON(i) - AbsDON(j)},
for any i and any j>i,
```

where i and j indicate the index of the NAL unit in the transmission order and AbsDON denotes a decoding order number of the NAL unit that does not wrap around to 0 after 65535. In other words, AbsDON is calculated as follows: let m and n be consecutive NAL units in transmission order. For the very first NAL unit in transmission order (whose index is 0), AbsDON(0) = DON(0). For other NAL units, AbsDON is calculated as follows:

```
If DON(m) == DON(n), AbsDON(n) = AbsDON(m)

If (DON(m) < DON(n) and DON(n) - DON(m) < 32768),
  AbsDON(n) = AbsDON(m) + DON(n) - DON(m)

If (DON(m) > DON(n) and DON(m) - DON(n) >= 32768),
  AbsDON(n) = AbsDON(m) + 65536 - DON(m) + DON(n)

If (DON(m) < DON(n) and DON(n) - DON(m) >= 32768),
  AbsDON(n) = AbsDON(m) - (DON(m) + 65536 - DON(n))

If (DON(m) > DON(n) and DON(m) - DON(n) < 32768),
  AbsDON(n) = AbsDON(m) - (DON(m) - DON(n))
```

where DON(i) is the decoding order number of the NAL unit having index i in the transmission order. The decoding order number is specified in Section 5.5.

Informative note: Receivers may use sprop-max-don-diff to trigger which NAL units in the receiver buffer can be passed to the decoder.

max-rcmd-nalu-size:
This parameter MAY be used to signal the capabilities of a receiver. The parameter MUST NOT be used for any other purposes. The value of the parameter indicates the largest NALU size in bytes that the receiver can handle efficiently. The parameter value is a recommendation, not a strict upper boundary. The sender MAY create larger NALUs but must be aware that the handling of these may come at a higher cost than NALUs conforming to the limitation.

The value of max-rcmd-nalu-size MUST be an integer in the range of 0 to 4294967295, inclusive. If this parameter is not specified, no known limitation to the NALU size exists. Senders still have to consider the MTU size available between the sender and the receiver and SHOULD run MTU discovery for this purpose.

This parameter is motivated by, for example, an IP to H.223 video telephony gateway, where NALUs smaller than the H.223 transport data unit will be more efficient. A gateway may terminate IP; thus, MTU discovery will normally not work beyond the gateway.

Informative note: Setting this parameter to a lower than necessary value may have a negative impact.

sar-understood:
This parameter MAY be used to indicate a receiver capability and nothing else. The parameter indicates the maximum value of aspect_ratio_idc (specified in [1]) smaller than 255 that the receiver understands. Table E-1 of [1] specifies aspect_ratio_idc equal to 0 as "unspecified"; 1 to 16, inclusive, as specific Sample Aspect Ratios (SARs); 17 to 254, inclusive, as "reserved"; and 255 as the Extended SAR, for which SAR width and SAR height are explicitly signaled. Therefore, a receiver with a decoder according to [1] understands aspect_ratio_idc in the range of 1 to 16, inclusive, and aspect_ratio_idc equal to 255, in the sense that the receiver knows exactly what the SAR is. For such a receiver, the value of sar-understood is 16. In the future, if Table E-1 of [1] is extended, e.g., such that the SAR for aspect_ratio_idc equal to 17 is specified, then for a receiver with a decoder that understands the extension, the value of

sar-understood is 17. For a receiver with a decoder according to the 2003 version of [1], the value of sar-understood is 13, as the minimum reserved aspect_ratio_idc therein is 14.

When sar-understood is not present, the value MUST be inferred to be equal to 13.

sar-supported:
This parameter MAY be used to indicate a receiver capability and nothing else. The value of this parameter is an integer in the range of 1 to sar-understood, inclusive, equal to 255. The value of sar-supported equal to N smaller than 255 indicates that the receiver supports all the SARs corresponding to H.264 aspect_ratio_idc values (see Table E-1 of [1]) in the range from 1 to N, inclusive, without geometric distortion. The value of sar-supported equal to 255 indicates that the receiver supports all sample aspect ratios that are expressible using two 16-bit integer values as the numerator and denominator, i.e., those that are expressible using the H.264 aspect_ratio_idc value of 255 (Extended_SAR, see Table E-1 of [1]), without geometric distortion.

H.264-compliant encoders SHOULD NOT send an aspect_ratio_idc equal to 0 or an aspect_ratio_idc larger than sar-understood and smaller than 255. H.264-compliant encoders SHOULD send an aspect_ratio_idc that the receiver is able to display without geometrical distortion. However, H.264-compliant encoders MAY choose to send pictures using any SAR.

Note that the actual sample aspect ratio or extended sample aspect ratio, when present, of the stream is conveyed in the Video Usability Information (VUI) part of the sequence parameter set.

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

Security considerations:
See Section 9 of RFC 6184.

Public specification:
Please refer to RFC 6184 and its Section 17.

Additional information:
None

File extensions: none

Macintosh file type code: none

Object identifier or OID: none

Person & email address to contact for further information:
Ye-Kui Wang, yekui.wang@huawei.com

Intended usage: COMMON

Author:
Ye-Kui Wang, yekui.wang@huawei.com

Change controller:
IETF Audio/Video Transport working group delegated from the IESG.

8.2. SDP Parameters

The receiver MUST ignore any parameter unspecified in this memo.

8.2.1. Mapping of Payload Type Parameters to SDP

The media type video/H264 string is mapped to fields in the Session Description Protocol (SDP) [6] as follows:

- The media name in the "m=" line of SDP MUST be video.
- The encoding name in the "a=rtpmap" line of SDP MUST be H264 (the media subtype).
- The clock rate in the "a=rtpmap" line MUST be 90000.
- The OPTIONAL parameters profile-level-id, max-recv-level, max-mbps, max-smbps, max-fs, max-cpb, max-dpb, max-br, redundant-pic-cap, use-level-src-parameter-sets, in-band-parameter-sets, level-asymmetry-allowed, packetization-mode, sprop-interleaving-depth, sprop-deint-buf-req, deint-buf-cap, sprop-init-buf-time, sprop-max-don-diff, max-rcmd-nalu-size, sar-understood, and sar-supported, when present, MUST be included in the "a=fmtp" line of SDP. These parameters are expressed as a media type string, in the form of a semicolon-separated list of parameter=value pairs.
- The OPTIONAL parameters sprop-parameter-sets and sprop-level-parameter-sets, when present, MUST be included in the "a=fmtp" line of SDP or conveyed using the "fmtp" source attribute as specified in Section 6.3 of [9]. For a particular media format (i.e., RTP payload type), a sprop-parameter-sets or sprop-level-parameter-sets MUST NOT be both included in the "a=fmtp" line of

SDP and conveyed using the "fmtp" source attribute. When included in the "a=fmtp" line of SDP, these parameters are expressed as a media type string, in the form of a semicolon-separated list of parameter=value pairs. When conveyed using the "fmtp" source attribute, these parameters are only associated with the given source and payload type as parts of the "fmtp" source attribute.

Informative note: Conveyance of sprop-parameter-sets and sprop-level-parameter-sets using the "fmtp" source attribute allows for out-of-band transport of parameter sets in topologies like Topo-Video-switch-MCU [29].

An example of media representation in SDP is as follows (Baseline profile, Level 3.0, some of the constraints of the Main profile may not be obeyed):

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E;
          packetization-mode=1;
          sprop-parameter-sets=<parameter sets data>
```

8.2.2. Usage with the SDP Offer/Answer Model

When H.264 is offered over RTP using SDP in an Offer/Answer model [8] for negotiation for unicast usage, the following limitations and rules apply:

- The parameters identifying a media format configuration for H.264 are profile-level-id and packetization-mode. These media format configuration parameters (except for the level part of profile-level-id) MUST be used symmetrically; that is, the answerer MUST either maintain all configuration parameters or remove the media format (payload type) completely if one or more of the parameter values are not supported. Note that the level part of profile-level-id includes level_idc, and, for indication of Level 1b when profile_idc is equal to 66, 77, or 88, bit 4 (constraint_set3_flag) of profile-iop. The level part of profile-level-id is changeable.

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

  Informative note: In H.264 [1], all the levels except for Level 1b are equal to the value of level_idc divided by 10. Level 1b is a level higher than Level 1.0 but lower than Level 1.1 and is signaled in an ad hoc manner, because the level was

specified after Level 1.0 and Level 1.1. For the Baseline, Main, and Extended profiles (with profile_idc equal to 66, 77, and 88, respectively), Level 1b is indicated by level_idc equal to 11 (i.e., same as Level 1.1) and constraint_set3_flag equal to 1. For other profiles, Level 1b is indicated by level_idc equal to 9 (but note that Level 1b for these profiles are still higher than Level 1, which has level_idc equal to 10 and lower than Level 1.1). In SDP Offer/Answer, an answer to an offer may indicate a level equal to or lower than the level indicated in the offer. Due to the ad hoc indication of Level 1b, offerers and answerers must check the value of bit 4 (constraint_set3_flag) of the middle octet of the parameter profile-level-id, when profile_idc is equal to 66, 77, or 88 and level_idc is equal to 11.

To simplify the handling and matching of these configurations, the same RTP payload type number used in the offer SHOULD also be used in the answer, as specified in [8]. An answer MUST NOT contain the payload type number used in the offer unless the configuration is exactly the same as in the offer.

Informative note: When an offerer receives an answer, it has to compare payload types not declared in the offer based on the media type (i.e., video/H264) and the above media configuration parameters with any payload types it has already declared. This will enable it to determine whether the configuration in question is new or if it is equivalent to configuration already offered, since a different payload type number may be used in the answer.

- When present, the parameter max-recv-level declares the highest level supported for receiving. In case max-recv-level is not present, the highest level supported for receiving is equal to the default level indicated by the level part of profile-level-id. When present, max-recv-level MUST be higher than the default level.

- The parameter level-asymmetry-allowed indicates whether level asymmetry is allowed.

  If level-asymmetry-allowed is equal to 0 (or not present) in either the offer or the answer, level asymmetry is not allowed. In this case, the level to use in the direction from the offerer to the answerer MUST be the same as the level to use in the opposite direction, and the common level to use is equal to the lower value of the default level in the offer and the default level in the answer.

Otherwise, level-asymmetry-allowed equals 1 in both the offer and the answer, and level asymmetry is allowed. In this case, the level to use in the offerer-to-answerer direction MUST be equal to the highest level the answerer supports for receiving, and the level to use in the answerer-to-offerer direction MUST be equal to the highest level the offerer supports for receiving.

When level asymmetry is not allowed, level upgrade is not allowed, i.e., the default level in the answer MUST be equal to or lower than the default level in the offer.

- The parameters sprop-deint-buf-req, sprop-interleaving-depth, sprop-max-don-diff, and sprop-init-buf-time describe the properties of the RTP packet stream that the offerer or answerer is sending for the media format configuration. This differs from the normal usage of the Offer/Answer parameters: normally such parameters declare the properties of the stream that the offerer or the answerer is able to receive. When dealing with H.264, the offerer assumes that the answerer will be able to receive media encoded using the configuration being offered.

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

- The capability parameters max-mbps, max-smbps, max-fs, max-cpb, max-dpb, max-br, redundant-pic-cap, max-rcmd-nalu-size, sar-understood, and sar-supported MAY be used to declare further capabilities of the offerer or answerer for receiving. These parameters MUST NOT be present when the direction attribute is "sendonly" and when the parameters describe the limitations of what the offerer or answerer accepts for receiving streams.

- An offerer has to include the size of the de-interleaving buffer, sprop-deint-buf-req, in the offer for an interleaved H.264 stream. To enable the offerer and answerer to inform each other about their capabilities for de-interleaving buffering in receiving streams, both parties are RECOMMENDED to include deint-buf-cap. For interleaved streams, it is also RECOMMENDED to consider offering multiple payload types with different buffering requirements when the capabilities of the receiver are unknown.

- The sprop-parameter-sets or sprop-level-parameter-sets parameter, when present (included in the "a=fmtp" line of SDP or conveyed using the "fmtp" source attribute as specified in Section 6.3 of

[9]), is used for out-of-band transport of parameter sets. However, when out-of-band transport of parameter sets is used, parameter sets MAY still be additionally transported in-band.

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

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

- An offer MAY include either or both of sprop-parameter-sets and sprop-level-parameter-sets. If neither sprop-parameter-sets nor sprop-level-parameter-sets is present in the offer, then only in-band transport of parameter sets is used.
- If the answer includes in-band-parameter-sets equal to 1, then the offerer MUST transmit parameter sets in-band. Otherwise, the following applies.
  - If the level to use in the offerer-to-answerer direction is equal to the default level in the offer, the following applies.

    When there is a sprop-parameter-sets included in the "a=fmtp" line in the offer, the answerer MUST be prepared to use the parameter sets included in the sprop-parameter-sets for decoding the incoming NAL unit stream.

    When there is a sprop-parameter-sets conveyed using the "fmtp" source attribute in the offer, the following applies. If the answer includes use-level-src-parameter-sets equal to 1 or the "fmtp" source attribute, the answerer MUST be prepared to use the parameter sets included in the sprop-parameter-sets for decoding the incoming NAL unit stream; otherwise, the offerer MUST transmit parameter sets in-band.

    When sprop-parameter-sets is not present in the offer, the offerer MUST transmit parameter sets in-band.

The answerer MUST ignore sprop-level-parameter-sets, when present (either included in the "a=fmtp" line or conveyed using the "fmtp" source attribute) in the offer.

- Otherwise, the level to use in the offerer-to-answerer direction is not equal to the default level in the offer, and the following applies.

  The answerer MUST ignore sprop-parameter-sets, when present (either included in the "a=fmtp" line or conveyed using the "fmtp" source attribute) in the offer.

  When neither use-level-src-parameter-sets is equal to 1 nor the "fmtp" source attribute is present in the answer, the answerer MUST ignore sprop-level-parameter-sets, when present in the offer, and the offerer MUST transmit parameter sets in-band.

  When either use-level-src-parameter-sets is equal to 1 or the "fmtp" source attribute is present in the answer, the answerer MUST be prepared to use the parameter sets that are included in sprop-level-parameter-sets for the accepted level (i.e., the default level in the answer), when present in the offer, for decoding the incoming NAL unit stream, and ignore all other parameter sets included in sprop-level-parameter-sets.

  When no parameter sets for the level to use in the offerer-to-answerer direction are present in sprop-level-parameter-sets in the offer, the offerer MUST transmit parameter sets in-band.

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

- An answer MAY include either sprop-parameter-sets or sprop-level-parameter-sets but MUST NOT include both. If neither sprop-parameter-sets nor sprop-level-parameter-sets is present in the answer, then only in-band transport of parameter sets is used.

- If the offer includes in-band-parameter-sets equal to 1, the answerer MUST NOT include sprop-parameter-sets or sprop-level-parameter-sets in the answer and MUST transmit parameter sets in-band. Otherwise, the following applies.

- If the level to use in the answerer-to-offerer direction is equal to the default level in the answer, the following applies.

  When there is a sprop-parameter-sets included in the "a=fmtp" line in the answer, the offerer MUST be prepared to use the parameter sets included in the sprop-parameter-sets for decoding the incoming NAL unit stream.

  When there is a sprop-parameter-sets conveyed using the "fmtp" source attribute in the answer, the following applies. If the offer includes use-level-src-parameter-sets equal to 1 or the "fmtp" source attribute, the offerer MUST be prepared to use the parameter sets included in the sprop-parameter-sets for decoding the incoming NAL unit stream; otherwise, the answerer MUST transmit parameter sets in-band.

  When sprop-parameter-sets is not present in the answer, the answerer MUST transmit parameter sets in-band.

  The offerer MUST ignore sprop-level-parameter-sets, when present (either included in the "a=fmtp" line or conveyed using the "fmtp" source attribute) in the answer.

- Otherwise, the level to use in the answerer-to-offerer direction is not equal to the default level in the answer, and the following applies.

  The offerer MUST ignore sprop-parameter-sets when present (either included in the "a=fmtp" line of SDP or conveyed using the "fmtp" source attribute) in the answer.

  When neither use-level-src-parameter-sets is equal to 1 nor the "fmtp" source attribute is present in the offer, the offerer MUST ignore sprop-level-parameter-sets, when present, and the answerer MUST transmit parameter sets in-band.

  When either use-level-src-parameter-sets is equal to 1 or the "fmtp" source attribute is present in the offer, the offerer MUST be prepared to use the parameter sets that are included in sprop-level-

parameter-sets for the level to use in the answerer-to-offerer direction, when present in the answer, for decoding the incoming NAL unit stream, and ignore all other parameter sets included in sprop-level-parameter-sets in the answer.

When no parameter sets for the level to use in the answerer-to-offerer direction are present in sprop-level-parameter-sets in the answer, the answerer MUST transmit parameter sets in-band.

When sprop-parameter-sets or sprop-level-parameter-sets is conveyed using the "fmtp" source attribute as specified in Section 6.3 of [9], the receiver of the parameters MUST store the parameter sets included in the sprop-parameter-sets or sprop-level-parameter-sets for the accepted level and associate them with the source given as a part of the "fmtp" source attribute. Parameter sets associated with one source MUST only be used to decode NAL units conveyed in RTP packets from the same source. When this mechanism is in use, SSRC collision detection and resolution MUST be performed as specified in [9].

Informative note: Conveyance of sprop-parameter-sets and sprop-level-parameter-sets using the "fmtp" source attribute may be used in topologies like Topo-Video-switch-MCU [29] to enable out-of-band transport of parameter sets.

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

- The media format configuration is identified by "profile-level-id", including the level part, and packetization-mode. These media format configuration parameters (including the level part of profile-level-id) MUST be used symmetrically; that is, the answerer MUST either maintain all configuration parameters or remove the media format (payload type) completely. Note that this implies that the level part of profile-level-id for Offer/Answer in multicast is not changeable.

  To simplify the handling and matching of these configurations, the same RTP payload type number used in the offer SHOULD also be used in the answer, as specified in [8]. An answer MUST NOT contain a payload type number used in the offer unless the configuration is the same as in the offer.

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

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

Table 6 lists the interpretation of all the media type parameters that MUST be used for the different direction attributes.

Table 6. Interpretation of parameters for different direction attributes

	sendrecv	recvonly	sendonly
profile-level-id	C	C	P
max-recv-level	R	R	-
packetization-mode	C	C	P
sprop-deint-buf-req	P	-	P
sprop-interleaving-depth	P	-	P
sprop-max-don-diff	P	-	P
sprop-init-buf-time	P	-	P
max-mbps	R	R	-
max-smbps	R	R	-
max-fs	R	R	-
max-cpb	R	R	-
max-dpb	R	R	-
max-br	R	R	-
redundant-pic-cap	R	R	-
deint-buf-cap	R	R	-
max-rcmd-nalu-size	R	R	-
sar-understood	R	R	-
sar-supported	R	R	-
in-band-parameter-sets	R	R	-
use-level-src-parameter-sets	R	R	-
level-asymmetry-allowed	O	-	-
sprop-parameter-sets	S	-	S
sprop-level-parameter-sets	S	-	S

Legend:

C: configuration for sending and receiving streams
O: offer/answer mode
P: properties of the stream to be sent
R: receiver capabilities
S: out-of-band parameter sets
-: not usable (when present, SHOULD be ignored)

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

Parameters declaring a configuration point are not changeable, with the exception of the level part of the profile-level-id parameter for unicast usage.

When a sender's capabilities are declared and non-downgradable parameters are used in this declaration, these parameters express a configuration that is acceptable for the sender to receive streams. In order to achieve high interoperability levels, it is often advisable to offer multiple alternative configurations, e.g., for the packetization mode. It is impossible to offer multiple configurations in a single payload type. Thus, when multiple configuration offers are made, each offer requires its own RTP payload type associated with the offer.

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

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

If an offerer wishes to have non-symmetric capabilities between sending and receiving, the offerer can allow asymmetric levels via level-asymmetry-allowed being equal to 1. Alternatively, the offerer could offer different RTP sessions, i.e., different media lines declared as "recvonly" and "sendonly", respectively. This may have further implications on the system and may require additional external semantics to associate the two media lines.

8.2.3. Usage in Declarative Session Descriptions

When H.264 over RTP is offered with SDP in a declarative style, as in Real Time Streaming Protocol (RTSP) [27] or Session Announcement Protocol (SAP) [28], the following considerations are necessary.

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

  Declaring actual configuration or stream properties:

  - profile-level-id
  - packetization-mode
  - sprop-interleaving-depth
  - sprop-deint-buf-req
  - sprop-max-don-diff
  - sprop-init-buf-time

  Out-of-band transporting of parameter sets:

  - sprop-parameter-sets
  - sprop-level-parameter-sets

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

  - max-mbps
  - max-smbps
  - max-fs
  - max-cpb
  - max-dpb
  - max-br
  - max-recv-level
  - redundant-pic-cap
  - max-rcmd-nalu-size
  - deint-buf-cap
  - sar-understood
  - sar-supported
  - in-band-parameter-sets
  - level-asymmetry-allowed
  - use-level-src-parameter-sets

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

8.3. Examples

An SDP Offer/Answer exchange wherein both parties are expected to both send and receive could look like the following. Only the media-codec-specific parts of the SDP are shown. Some lines are wrapped due to text constraints.

Offerer -> Answerer SDP message:

```
m=video 49170 RTP/AVP 100 99 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E; packetization-mode=0;
  sprop-parameter-sets=<parameter sets data#0>
a=rtpmap:99 H264/90000
a=fmtp:99 profile-level-id=42A01E; packetization-mode=1;
  sprop-parameter-sets=<parameter sets data#1>
a=rtpmap:100 H264/90000
a=fmtp:100 profile-level-id=42A01E; packetization-mode=2;
  sprop-parameter-sets=<parameter sets data#2>;
  sprop-interleaving-depth=45; sprop-deint-buf-req=64000;
  sprop-init-buf-time=102478; deint-buf-cap=128000
```

The above offer presents the same codec configuration in three different packetization formats. Payload type 98 represents single NALU mode, payload type 99 represents non-interleaved mode, and payload type 100 indicates the interleaved mode. In the interleaved mode case, the interleaving parameters that the offerer would use if the answer indicates support for payload type 100 are also included. In all three cases, the parameter sprop-parameter-sets conveys the initial parameter sets that are required by the answerer when receiving a stream from the offerer when this configuration is accepted. Note that the value for sprop-parameter-sets could be different for each payload type.

```
Answerer -> Offerer SDP message:

m=video 49170 RTP/AVP 100 99 97
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42A01E; packetization-mode=0;
  sprop-parameter-sets=<parameter sets data#3>
a=rtpmap:99 H264/90000
a=fmtp:99 profile-level-id=42A01E; packetization-mode=1;
  sprop-parameter-sets=<parameter sets data#4>;
  max-rcmd-nalu-size=3980
a=rtpmap:100 H264/90000
a=fmtp:100 profile-level-id=42A01E; packetization-mode=2;
  sprop-parameter-sets=<parameter sets data#5>;
  sprop-interleaving-depth=60;
  sprop-deint-buf-req=86000; sprop-init-buf-time=156320;
  deint-buf-cap=128000; max-rcmd-nalu-size=3980
```

As the Offer/Answer negotiation covers both sending and receiving streams, an offer indicates the exact parameters for what the offerer is willing to receive, whereas the answer indicates the same for what the answerer is willing to receive. In this case, the offerer declared that it is willing to receive payload type 98. The answerer accepts this by declaring an equivalent payload type 97; that is, it has identical values for the two parameters profile-level-id and packetization-mode (since packetization-mode is equal to 0 and sprop-deint-buf-req is not present). As the offered payload type 98 is accepted, the answerer needs to store parameter sets included in sprop-parameter-sets=<parameter sets data#0> in case the offer finally decides to use this configuration. In the answer, the answerer includes the parameter sets in sprop-parameter-sets=<parameter sets data#3> that the answerer would use in the stream sent from the answerer if this configuration is finally used.

The answerer also accepts the reception of the two configurations that payload types 99 and 100 represent. Again, the answerer needs to store parameter sets included in sprop-parameter-sets=<parameter sets data#1> and sprop-parameter-sets=<parameter sets data#2> in case the offer finally decides to use either of these two configurations. The answerer provides the initial parameter sets for the answerer-to-offerer direction, i.e., the parameter sets in sprop-parameter-sets=<parameter sets data#4> and sprop-parameter-sets=<parameter sets data#5>, for payload types 99 and 100, respectively, that it will use to send the payload types. The answerer also provides the offerer with its memory limit for de-interleaving operations by providing a deint-buf-cap parameter. This is only useful if the offerer decides on making a second offer, where it can take the new value into

account. The max-rcmd-nalu-size indicates that the answerer can efficiently process NALUs up to the size of 3980 bytes. However, there is no guarantee that the network supports this size.

In the following example, the offer is accepted without level downgrading (i.e., the default level, Level 3.0, is accepted), and both sprop-parameter-sets and sprop-level-parameter-sets are present in the offer. The answerer must ignore sprop-level-parameter-sets=<parameter sets data#1> and store parameter sets in sprop-parameter-sets=<parameter sets data#0> for decoding the incoming NAL unit stream. The offerer must store the parameter sets in sprop-parameter-sets=<parameter sets data#2> in the answer for decoding the incoming NAL unit stream. Note that in this example, parameter sets in sprop-parameter-sets=<parameter sets data#2> must be associated with Level 3.0.

Offer SDP:

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E; //Baseline profile, Level 3.0
  packetization-mode=1;
  sprop-parameter-sets=<parameter sets data#0>;
  sprop-level-parameter-sets=<parameter sets data#1>
```

Answer SDP:

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E; //Baseline profile, Level 3.0
  packetization-mode=1;
  sprop-parameter-sets=<parameter sets data#2>
```

In the following example, the offer (Baseline profile, Level 1.1) is accepted with level downgrading (the accepted level is Level 1b), and both sprop-parameter-sets and sprop-level-parameter-sets are present in the offer. The answerer must ignore sprop-parameter-sets=<parameter sets data#0> and all parameter sets not for the accepted level (Level 1b) in sprop-level-parameter-sets=<parameter sets data#1> and must store parameter sets for the accepted level (Level 1b) in sprop-level-parameter-sets=<parameter sets data#1> for decoding the incoming NAL unit stream. The offerer must store the parameter sets in sprop-parameter-sets=<parameter sets data#2> in the answer for decoding the incoming NAL unit stream. Note that in this example, parameter sets in sprop-parameter-sets=<parameter sets data#2> must be associated with Level 1b.

```
Offer SDP:

m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A00B; //Baseline profile, Level 1.1
  packetization-mode=1;
  sprop-parameter-sets=<parameter sets data#0>;
  sprop-level-parameter-sets=<parameter sets data#1>

Answer SDP:

m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42B00B; //Baseline profile, Level 1b
  packetization-mode=1;
  sprop-parameter-sets=<parameter sets data#2>;
  use-level-src-parameter-sets=1
```

In the following example, the offer (Baseline profile, Level 1.1) is accepted with level downgrading (the accepted level is Level 1b), and both sprop-parameter-sets and sprop-level-parameter-sets are present in the offer. However, the answerer is a legacy RFC 3984 implementation and does not understand sprop-level-parameter-sets; hence, it does not include use-level-src-parameter-sets (which the answerer does not understand either) in the answer. Therefore, the answerer must ignore both sprop-parameter-sets=<parameter sets data#0> and sprop-level-parameter-sets=<parameter sets data#1>, and the offerer must transport parameter sets in-band.

```
Offer SDP:

m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A00B; //Baseline profile, Level 1.1
  packetization-mode=1;
  sprop-parameter-sets=<parameter sets data#0>;
  sprop-level-parameter-sets=<parameter sets data#1>

Answer SDP:

m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42B00B; //Baseline profile, Level 1b
  packetization-mode=1
```

In the following example, the offer is accepted without level downgrading, and sprop-parameter-sets is present in the offer. Parameter sets in sprop-parameter-sets=<parameter sets data#0> must

be stored and used by the encoder of the offerer and the decoder of the answerer, and parameter sets in sprop-parameter-sets=<parameter sets data#1> must be used by the encoder of the answerer and the decoder of the offerer. Note that sprop-parameter-sets=<parameter sets data#0> is basically independent of sprop-parameter-sets=<parameter sets data#1>.

Offer SDP:

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E; //Baseline profile, Level 3.0
  packetization-mode=1;
  sprop-parameter-sets=<parameter sets data#0>
```

Answer SDP:

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E; //Baseline profile, Level 3.0
  packetization-mode=1;
  sprop-parameter-sets=<parameter sets data#1>
```

In the following example, the offer is accepted without level downgrading, and neither sprop-parameter-sets nor sprop-level-parameter-sets is present in the offer, meaning that there is no out-of-band transmission of parameter sets, which then have to be transported in-band.

Offer SDP:

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E; //Baseline profile, Level 3.0
  packetization-mode=1
```

Answer SDP:

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E; //Baseline profile, Level 3.0
  packetization-mode=1
```

In the following example, the offer is accepted with level downgrading and sprop-parameter-sets is present in the offer. As sprop-parameter-sets=<parameter sets data#0> contains level_idc indicating Level 3.0, it therefore cannot be used, as the answerer wants Level 2.0, and must be ignored by the answerer, and in-band parameter sets must be used.

Offer SDP:

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E; //Baseline profile, Level 3.0
  packetization-mode=1;
  sprop-parameter-sets=<parameter sets data#0>
```

Answer SDP:

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A014; //Baseline profile, Level 2.0
  packetization-mode=1
```

In the following example, the offer is also accepted with level downgrading, and neither sprop-parameter-sets nor sprop-level-parameter-sets is present in the offer, meaning that there is no out-of-band transmission of parameter sets, which then have to be transported in-band.

Offer SDP:

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E; //Baseline profile, Level 3.0
  packetization-mode=1
```

Answer SDP:

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A014; //Baseline profile, Level 2.0
  packetization-mode=1
```

In the following example, the offer is accepted with level upgrading, and neither sprop-parameter-sets nor sprop-level-parameter-sets is present in the offer or the answer, meaning that there is no out-of-band transmission of parameter sets, which then have to be transported in-band. The level to use in the offerer-to-answerer direction is Level 3.0, and the level to use in the answerer-to-

offerer direction is Level 2.0. The answerer is allowed to send at any level up to and including Level 2.0, and the offerer is allowed to send at any level up to and including Level 3.0.

Offer SDP:

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A014; //Baseline profile, Level 2.0
  packetization-mode=1; level-asymmetry-allowed=1
```

Answer SDP:

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E; //Baseline profile, Level 3.0
  packetization-mode=1; level-asymmetry-allowed=1
```

In the following example, the offerer is a Multipoint Control Unit (MCU) in a topology like Topo-Video-switch-MCU [29], offering parameter sets received (using out-of-band transport) from three other participants (B, C, and D) and receiving parameter sets from the participant A, which is the answerer. The participants are identified by their values of canonical name (CNAME), which are mapped to different SSRC values. The same codec configuration is used by all four participants. The participant A stores and associates the parameter sets included in <parameter sets data#B>, <parameter sets data#C>, and <parameter sets data#D> to participants B, C, and D, respectively, and uses <parameter sets data#B> for decoding NAL units carried in RTP packets originating from participant B only, uses <parameter sets data#C> for decoding NAL units carried in RTP packets originating from participant C only, and uses <parameter sets data#D> for decoding NAL units carried in RTP packets originating from participant D only.

```
Offer SDP:

m=video 49170 RTP/AVP 98
a=ssrc:SSRC-B cname:CNAME-B
a=ssrc:SSRC-C cname:CNAME-C
a=ssrc:SSRC-D cname:CNAME-D
a=ssrc:SSRC-B fmtp:98
  sprop-parameter-sets=<parameter sets data#B>
a=ssrc:SSRC-C fmtp:98
  sprop-parameter-sets=<parameter sets data#C>
a=ssrc:SSRC-D fmtp:98
  sprop-parameter-sets=<parameter sets data#D>
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E; //Baseline profile, Level 3.0
  packetization-mode=1

Answer SDP:

m=video 49170 RTP/AVP 98
a=ssrc:SSRC-A cname:CNAME-A
a=ssrc:SSRC-A fmtp:98
  sprop-parameter-sets=<parameter sets data#A>
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E; //Baseline profile, Level 3.0
  packetization-mode=1
```

8.4. Parameter Set Considerations

The H.264 parameter sets are a fundamental part of the video codec and vital to its operation (see Section 1.2). Due to their characteristics and their importance for the decoding process, lost or erroneously transmitted parameter sets can hardly be concealed locally at the receiver. A reference to a corrupt parameter set normally has fatal results to the decoding process. Corruption could occur, for example, due to the erroneous transmission or loss of a parameter set NAL unit but also due to the untimely transmission of a parameter set update. A parameter set update refers to a change of at least one parameter in a picture parameter set or sequence parameter set for which the picture parameter set or sequence parameter set identifier remains unchanged. Therefore, the following recommendations are provided as a guideline for the implementer of the RTP sender.

Parameter set NALUs can be transported using three different principles:

A. Using a session control protocol (out-of-band) prior to the actual RTP session.

B. Using a session control protocol (out-of-band) during an ongoing RTP session.

C. Within the RTP packet stream in the payload (in-band) during an ongoing RTP session.

It is recommended to implement principles A and B within a session control protocol. SIP and SDP can be used as described in the SDP Offer/Answer model and in the previous sections of this memo. Section 8.2.2 includes a detailed discussion on transport of parameter sets in-band or out-of-band in SDP Offer/Answer using media type parameters sprop-parameter-sets, sprop-level-parameter-sets, use-level-src-parameter-sets, and in-band-parameter-sets. This section contains guidelines on how principles A and B should be implemented within session control protocols. It is independent of the particular protocol used. Principle C is supported by the RTP payload format defined in this specification. There are topologies like Topo-Video-switch-MCU [29] for which the use of principle C may be desirable.

If in-band signaling of parameter sets is used, the picture and sequence parameter set NALUs SHOULD be transmitted in the RTP payload using a reliable method of delivering of RTP (see below), as a loss of a parameter set of either type will likely prevent decoding of a considerable portion of the corresponding RTP packet stream.

If in-band signaling of parameter sets is used, the sender SHOULD take the error characteristics into account and use mechanisms to provide a high probability for delivering the parameter sets correctly. Mechanisms that increase the probability for a correct reception include packet repetition, FEC, and retransmission. The use of an unreliable, out-of-band control protocol has similar disadvantages as the in-band signaling (possible loss) and, in addition, may also lead to difficulties in the synchronization (see below). Therefore, it is NOT RECOMMENDED.

Parameter sets MAY be added or updated during the lifetime of a session using principles B and C. It is required that parameter sets be present at the decoder prior to the NAL units that refer to them. Update or addition of parameter sets can result in further problems; therefore, the following recommendations should be considered.

- When parameter sets are added or updated, care SHOULD be taken to ensure that any parameter set is delivered prior to its usage. When new parameter sets are added, previously unused parameter set identifiers are used. It is common that no synchronization is present between out-of-band signaling and in-band traffic. If out-of-band signaling is used, it is RECOMMENDED that a sender not start sending NALUs requiring the added or updated parameter sets prior to acknowledgement of delivery from the signaling protocol.

- When parameter sets are updated, the following synchronization issue should be taken into account. When overwriting a parameter set at the receiver, the sender has to ensure that the parameter set in question is not needed by any NALU present in the network or receiver buffers. Otherwise, decoding with a wrong parameter set may occur. To lessen this problem, it is RECOMMENDED either to overwrite only those parameter sets that have not been used for a sufficiently long time (to ensure that all related NALUs have been consumed) or to add a new parameter set instead (which may have negative consequences for the efficiency of the video coding).

  Informative note: In some topologies like Topo-Video-switch-MCU [29], the origin of the whole set of parameter sets may come from multiple sources that may use non-unique parameter set identifiers. In this case, an offer may overwrite an existing parameter set if no other mechanism that enables uniqueness of the parameter sets in the out-of-band channel exists.

- In a multiparty session, one participant MUST associate parameter sets coming from different sources with the source identification whenever possible, e.g., by conveying out-of-band transported parameter sets, as different sources typically use independent parameter set identifier value spaces.

- Adding or modifying parameter sets by using both principles B and C in the same RTP session may lead to inconsistencies of the parameter sets because of the lack of synchronization between the control and the RTP channel. Therefore, principles B and C MUST NOT both be used in the same session unless sufficient synchronization can be provided.

In some scenarios (e.g., when only the subset of this payload format specification corresponding to H.241 is used) or topologies, it is not possible to employ out-of-band parameter set transmission. In this case, parameter sets have to be transmitted in-band. Here, the synchronization with the non-parameter-set-data in the bitstream is implicit, but the possibility of a loss has to be taken into account.

The loss probability should be reduced using the mechanisms discussed above. In case a loss of a parameter set is detected, recovery may be achieved using a Decoder Refresh Point procedure, for example, using RTCP feedback Full Intra Request (FIR) [30]. Two example Decoder Refresh Point procedures are provided in the informative Section 8.5.

- When parameter sets are initially provided using principle A and then later added or updated in-band (principle C), there is a risk associated with updating the parameter sets delivered out-of-band. If receivers miss some in-band updates (for example, because of a loss or a late tune-in), those receivers attempt to decode the bitstream using outdated parameters. It is therefore RECOMMENDED that parameter set IDs be partitioned between the out-of-band and in-band parameter sets.

8.5. Decoder Refresh Point Procedure Using In-Band Transport of Parameter Sets (Informative)

When a sender with a video encoder according to [1] receives a request for a decoder refresh point, the encoder shall enter the fast update mode by using one of the procedures specified in Sections 8.5.1 or 8.5.2. The procedure in Section 8.5.1 is the preferred response in a lossless transmission environment. Both procedures satisfy the requirement to enter the fast update mode for H.264 video encoding.

8.5.1. IDR Procedure to Respond to a Request for a Decoder Refresh Point

This section gives one possible way to respond to a request for a decoder refresh point.

The encoder shall, in the order presented here:

1) Immediately prepare to send an IDR picture.

2) Send a sequence parameter set to be used by the IDR picture to be sent. The encoder may optionally also send other sequence parameter sets.

3) Send a picture parameter set to be used by the IDR picture to be sent. The encoder may optionally also send other picture parameter sets.

4) Send the IDR picture.

5) From this point forward in time, send any other sequence or picture parameter sets that have not yet been sent in this procedure, prior to their reference by any NAL unit, regardless of whether such parameter sets were previously sent prior to receiving the request for a decoder refresh point. As needed, such parameter sets may be sent in a batch, one at a time, or in any combination of these two methods. Parameter sets may be re-sent at any time for redundancy. Caution should be taken when parameter set updates are present, as described above in Section 8.4.

8.5.2. Gradual Recovery Procedure to Respond to a Request for a Decoder Refresh Point

This section gives another possible way to respond to a request for a decoder refresh point.

The encoder shall, in the order presented here:

1) Send a recovery point SEI message (see Sections D.1.7 and D.2.7 of [1]).

2) Repeat any sequence and picture parameter sets that were sent before the recovery point SEI message, prior to their reference by a NAL unit.

The encoder shall ensure that the decoder has access to all reference pictures for inter prediction of pictures at or after the recovery point, which is indicated by the recovery point SEI message, in output order, assuming that the transmission from now on is error-free.

The value of the recovery_frame_cnt syntax element in the recovery point SEI message should be small enough to ensure a fast recovery.

As needed, such parameter sets may be re-sent in a batch, one at a time, or in any combination of these two methods. Parameter sets may be re-sent at any time for redundancy. Caution should be taken when parameter set updates are present, as described above in Section 8.4.

9. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [5] and in any appropriate RTP profile (for example, [16]). This implies that confidentiality of the media streams is achieved by encryption, for example, through the application of SRTP [26]. Because the data compression used with this payload format is

applied end-to-end, any encryption needs to be performed after compression. A potential denial-of-service threat exists for data encodings using compression techniques that have non-uniform receiver-end computational load. The attacker can inject pathological datagrams into the stream that are complex to decode and that cause the receiver to be overloaded. H.264 is particularly vulnerable to such attacks, as it is extremely simple to generate datagrams containing NAL units that affect the decoding process of many future NAL units. Therefore, the usage of data origin authentication and data integrity protection of at least the RTP packet is RECOMMENDED, for example, with SRTP [26].

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

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

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

10. Congestion Control

Congestion control for RTP SHALL be used in accordance with RFC 3550 [5] and with any applicable RTP profile, e.g., RFC 3551 [16]. If best-effort service is being used, an additional requirement is that users of this payload format MUST monitor packet loss to ensure that the packet loss rate is within acceptable parameters. Packet loss is considered acceptable if a TCP flow across the same network path, and experiencing the same network conditions, would achieve an average throughput, measured on a reasonable timescale, that is not less than the RTP flow is achieving. This condition can be satisfied by implementing congestion control mechanisms to adapt the transmission rate (or the number of layers subscribed for a layered multicast session) or by arranging for a receiver to leave the session if the loss rate is unacceptably high.

The bitrate adaptation necessary for obeying the congestion control principle is easily achievable when real-time encoding is used. However, when pre-encoded content is being transmitted, bandwidth adaptation requires the availability of more than one coded representation of the same content, at different bitrates, or the existence of non-reference pictures or sub-sequences [22] in the bitstream. The switching between the different representations can normally be performed in the same RTP session, e.g., by employing a concept known as SI/SP slices of the Extended profile or by switching streams at IDR picture boundaries. Only when non-downgradable parameters (such as the profile part of the profile/level ID) are required to be changed does it become necessary to terminate and restart the media stream. This may be accomplished by using a different RTP payload type.

MANEs MAY follow the suggestions outlined in Section 7.3 and remove certain unusable packets from the packet stream when that stream was damaged due to previous packet losses. This can help reduce the network load in certain special cases.

11. IANA Considerations

The H264 media subtype name specified by RFC 3984 has been updated as defined in Section 8.1 of this memo.

12. Informative Appendix: Application Examples

This payload specification is very flexible in its use, in order to cover the extremely wide application space anticipated for H.264. However, this great flexibility also makes it difficult for an implementer to decide on a reasonable packetization scheme. Some information on how to apply this specification to real-world scenarios is likely to appear in the form of academic publications and a test model software and description in the near future. However, some preliminary usage scenarios are described here as well.

12.1. Video Telephony According to Annex A of ITU-T Recommendation H.241

H.323-based video telephony systems that use H.264 as an optional video compression scheme are required to support Annex A of H.241 [3] as a packetization scheme. The packetization mechanism defined in this Annex is technically identical with a small subset of this specification.

When a system operates according to Annex A of H.241, parameter set NAL units are sent in-band. Only single NAL unit packets are used. Many such systems are not sending IDR pictures regularly, but only

when required by user interaction or by control protocol means, e.g., when switching between video channels in a Multipoint Control Unit or for error recovery requested by feedback.

12.2. Video Telephony, No Slice Data Partitioning, No NAL Unit Aggregation

The RTP part of this scheme is implemented and tested (though not the control-protocol part; see below).

In most real-world video telephony applications, picture parameters such as picture size or optional modes never change during the lifetime of a connection. Therefore, all necessary parameter sets (usually only one) are sent as a side effect of the capability exchange/announcement process, e.g., according to the SDP syntax specified in Section 8.2 of this document. As all necessary parameter set information is established before the RTP session starts, there is no need for sending any parameter set NAL units. Slice data partitioning is not used either. Thus, the RTP packet stream basically consists of NAL units that carry single coded slices.

The encoder chooses the size of coded slice NAL units so that they offer the best performance. Often, this is done by adapting the coded slice size to the MTU size of the IP network. For small picture sizes, this may result in a one-picture-per-one-packet strategy. Intra refresh algorithms clean up the loss of packets and the resulting drift-related artifacts.

12.3. Video Telephony, Interleaved Packetization Using NAL Unit Aggregation

This scheme allows better error concealment and is used in H.263-based designs using RFC 4629 packetization [11]. It has been implemented, and good results were reported [13].

The VCL encoder codes the source picture so that all macroblocks (MBs) of one MB line are assigned to one slice. All slices with even MB row addresses are combined into one STAP, and all slices with odd MB row addresses are combined into another. Those STAPs are transmitted as RTP packets. The establishment of the parameter sets is performed as discussed above.

Note that the use of STAPs is essential here, as the high number of individual slices (18 for a Common Intermediate Format (CIF) picture) would lead to unacceptably high IP/UDP/RTP header overhead (unless the source coding tool FMO is used, which is not assumed in this scenario). Furthermore, some wireless video transmission systems,

such as H.324M and the IP-based video telephony specified in 3GPP, are likely to use relatively small transport packet size. For example, a typical MTU size of H.223 AL3 SDU is around 100 bytes [17]. Coding individual slices according to this packetization scheme provides further advantage in communication between wired and wireless networks, as individual slices are likely to be smaller than the preferred maximum packet size of wireless systems. Consequently, a gateway can convert the STAPs used in a wired network into several RTP packets with only one NAL unit, which are preferred in a wireless network, and vice versa.

12.4. Video Telephony with Data Partitioning

This scheme has been implemented and has been shown to offer good performance, especially at higher packet loss rates [13].

Data partitioning is known to be useful only when some form of unequal error protection is available. Normally, in single-session RTP environments, even error characteristics are assumed; that is, the packet loss probability of all packets of the session is the same statistically. However, there are means to reduce the packet loss probability of individual packets in an RTP session. A FEC packet according to RFC 5109 [18], for example, specifies which media packets are associated with the FEC packet.

In all cases, the incurred overhead is substantial but is in the same order of magnitude as the number of bits that have otherwise been spent for intra information. However, this mechanism does not add any delay to the system.

Again, the complete parameter set establishment is performed through control protocol means.

12.5. Video Telephony or Streaming with FUs and Forward Error Correction

This scheme has been implemented and has been shown to provide good performance, especially at higher packet loss rates [19].

The most efficient means to combat packet losses for scenarios where retransmissions are not applicable is forward error correction (FEC). Although application layer, end-to-end use of FEC is often less efficient than a FEC-based protection of individual links (especially when links of different characteristics are in the transmission path), application layer, end-to-end FEC is unavoidable in some scenarios. RFC 5109 [18] provides means to use generic, application layer, end-to-end FEC in packet loss environments. A binary forward error correcting code is generated by applying the XOR operation to

the bits at the same bit position in different packets. The binary code can be specified by the parameters (n,k), in which k is the number of information packets used in the connection and n is the total number of packets generated for k information packets; that is, n-k parity packets are generated for k information packets.

When a code is used with parameters (n,k) within the RFC 5109 framework, the following properties are well known:

a) If applied over one RTP packet, RFC 5109 provides only packet repetition.

b) RFC 5109 is most bitrate efficient if XOR-connected packets have equal length.

c) At the same packet loss probability p and for a fixed k, the greater the value of n, the smaller the residual error probability becomes. For example, for a packet loss probability of 10%, k=1, and n=2, the residual error probability is about 1%, whereas for n=3, the residual error probability is about 0.1%.

d) At the same packet loss probability p and for a fixed code rate k/n, the greater the value of n, the smaller the residual error probability becomes. For example, at a packet loss probability of p=10%, k=1, and n=2, the residual error rate is about 1%, whereas for an extended Golay code with k=12 and n=24, the residual error rate is about 0.01%.

For applying RFC 5109 in combination with H.264 baseline-coded video without using FUs, several options might be considered:

1) The video encoder produces NAL units for which each video frame is coded in a single slice. Applying FEC, one could use a simple code, e.g., (n=2, k=1). That is, each NAL unit would basically just be repeated. The disadvantage is obviously the bad code performance according to d), above, and the low flexibility, as only (n, k=1) codes can be used.

2) The video encoder produces NAL units for which each video frame is encoded in one or more consecutive slices. Applying FEC, one could use a better code, e.g., (n=24, k=12), over a sequence of NAL units. Depending on the number of RTP packets per frame, a loss may introduce a significant delay, which is reduced when more RTP packets are used per frame. Packets of completely different lengths might also be connected, which decreases bitrate

efficiency according to b), above. However, with some care and for slices of 1 kb or larger, similar length (100-200 bytes difference) may be produced, which will not lower the bit efficiency catastrophically.

3) The video encoder produces NAL units, for which a certain frame contains k slices of possibly almost equal length. Then, applying FEC, a better code, e.g., (n=24, k=12), can be used over the sequence of NAL units for each frame. The delay compared to that of 2), above, may be reduced, but several disadvantages are obvious. First, the coding efficiency of the encoded video is lowered significantly, as slice-structured coding reduces intra-frame prediction and additional slice overhead is necessary. Second, pre-encoded content or, when operating over a gateway, the video is usually not appropriately coded with k slices such that FEC can be applied. Finally, the encoding of video producing k slices of equal length is not straightforward and might require more than one encoding pass.

Many of the mentioned disadvantages can be avoided by applying FUs in combination with FEC. Each NAL unit can be split into any number of FUs of basically equal length; therefore, FEC, with a reasonable k and n, can be applied, even if the encoder made no effort to produce slices of equal length. For example, a coded slice NAL unit containing an entire frame can be split to k FUs, and a parity check code (n=k+1, k) can be applied. However, this has the disadvantage that unless all created fragments can be recovered, the whole slice will be lost. Thus, a larger section is lost than would be if the frame had been split into several slices.

The presented technique makes it possible to achieve good transmission error tolerance, even if no additional source coding layer redundancy (such as periodic intra frames) is present. Consequently, the same coded video sequence can be used to achieve the maximum compression efficiency and quality over error-free transmission and for transmission over error-prone networks. Furthermore, the technique allows the application of FEC to pre-encoded sequences without adding delay. In this case, pre-encoded sequences that are not encoded for error-prone networks can still be transmitted almost reliably without adding extensive delays. In addition, FUs of equal length result in a bitrate efficient use of RFC 5109.

If the error probability depends on the length of the transmitted packet (e.g., in case of mobile transmission [15]), the benefits of applying FUs with FEC are even more obvious. Basically, the flexibility of the size of FUs allows appropriate FEC to be applied for each NAL unit and unequal error protection of NAL units.

When FUs and FEC are used, the incurred overhead is substantial but is in the same order of magnitude as the number of bits that have to be spent for intra-coded macroblocks if no FEC is applied. In [19], it was shown that the overall performance of the FEC-based approach enhanced quality when using the same error rate and same overall bitrate, including the overhead.

12.6. Low Bitrate Streaming

This scheme has been implemented with H.263 and non-standard RTP packetization and has given good results [20]. There is no technical reason why similarly good results could not be achievable with H.264.

In today's Internet streaming, some of the offered bitrates are relatively low in order to allow terminals with dial-up modems to access the content. In wired IP networks, relatively large packets, say 500 - 1500 bytes, are preferred to smaller and more frequently occurring packets in order to reduce network congestion. Moreover, use of large packets decreases the amount of RTP/UDP/IP header overhead. For low bitrate video, the use of large packets means that sometimes up to few pictures should be encapsulated in one packet.

However, the loss of a packet including many coded pictures would have drastic consequences for visual quality, as there is practically no way to conceal the loss of an entire picture other than repeating the previous one. One way to construct relatively large packets and maintain possibilities for successful loss concealment is to construct MTAPs that contain interleaved slices from several pictures. An MTAP should not contain spatially adjacent slices from the same picture or spatially overlapping slices from any picture. If a packet is lost, it is likely that a lost slice is surrounded by spatially adjacent slices of the same picture and spatially corresponding slices of the temporally previous and succeeding pictures. Consequently, concealment of the lost slice is likely to be relatively successful.

12.7. Robust Packet Scheduling in Video Streaming

Robust packet scheduling has been implemented with MPEG-4 Part 2 and simulated in a wireless streaming environment [21]. There is no technical reason why similar or better results could not be achievable with H.264.

Streaming clients typically have a receiver buffer that is capable of storing a relatively large amount of data. Initially, when a streaming session is established, a client does not start playing the stream back immediately. Rather, it typically buffers the incoming data for a few seconds. This buffering helps maintain continuous

playback, as, in case of occasional increased transmission delays or network throughput drops, the client can decode and play buffered data. Otherwise, without initial buffering, the client has to freeze the display, stop decoding, and wait for incoming data. The buffering is also necessary for either automatic or selective retransmission in any protocol level. If any part of a picture is lost, a retransmission mechanism may be used to resend the lost data. If the retransmitted data is received before its scheduled decoding or playback time, the loss is recovered perfectly. Coded pictures can be ranked according to their importance in the subjective quality of the decoded sequence. For example, non-reference pictures, such as conventional B pictures, are subjectively least important, as their absence does not affect decoding of any other pictures. In addition to non-reference pictures, the ITU-T H.264 | ISO/IEC 14496-10 standard includes a temporal scalability method called sub-sequences [22]. Subjective ranking can also be made on coded slice data partition or slice group basis. Coded slices and coded slice data partitions that are subjectively the most important can be sent earlier than their decoding order indicates, whereas coded slices and coded slice data partitions that are subjectively the least important can be sent later than their natural coding order indicates. Consequently, any retransmitted parts of the most important slices and coded slice data partitions are more likely to be received before their scheduled decoding or playback time compared to the least important slices and slice data partitions.

13. Informative Appendix: Rationale for Decoding Order Number

13.1. Introduction

The Decoding Order Number (DON) concept was introduced mainly to enable efficient multi-picture slice interleaving (see Section 12.6) and robust packet scheduling (see Section 12.7). In both of these applications, NAL units are transmitted out of decoding order. DON indicates the decoding order of NAL units and should be used in the receiver to recover the decoding order. Example use cases for efficient multi-picture slice interleaving and for robust packet scheduling are given in Sections 13.2 and 13.3, respectively. Section 13.4 describes the benefits of the DON concept in error resiliency achieved by redundant coded pictures. Section 13.5 summarizes considered alternatives to DON and justifies why DON was chosen for this RTP payload specification.

13.2. Example of Multi-Picture Slice Interleaving

An example of multi-picture slice interleaving follows. A subset of a coded video sequence is depicted below in output order. R denotes a reference picture, N denotes a non-reference picture, and the number indicates a relative output time.

```
... R1 N2 R3 N4 R5 ...
```

The decoding order of these pictures from left to right is as follows:

```
... R1 R3 N2 R5 N4 ...
```

The NAL units of pictures R1, R3, N2, R5, and N4 are marked with a DON equal to 1, 2, 3, 4, and 5, respectively.

Each reference picture consists of three slice groups that are scattered as follows (a number denotes the slice group number for each macroblock in a Quarter Common Intermediate Format (QCIF) frame):

```
0 1 2 0 1 2 0 1 2 0 1
2 0 1 2 0 1 2 0 1 2 0
1 2 0 1 2 0 1 2 0 1 2
0 1 2 0 1 2 0 1 2 0 1
2 0 1 2 0 1 2 0 1 2 0
1 2 0 1 2 0 1 2 0 1 2
0 1 2 0 1 2 0 1 2 0 1
2 0 1 2 0 1 2 0 1 2 0
1 2 0 1 2 0 1 2 0 1 2
```

For the sake of simplicity, we assume that all the macroblocks of a slice group are included in one slice. Three MTAPs are constructed from three consecutive reference pictures so that each MTAP contains three aggregation units, each of which contains all the macroblocks from one slice group. The first MTAP contains slice group 0 of picture R1, slice group 1 of picture R3, and slice group 2 of picture R5. The second MTAP contains slice group 1 of picture R1, slice group 2 of picture R3, and slice group 0 of picture R5. The third MTAP contains slice group 2 of picture R1, slice group 0 of picture R3, and slice group 1 of picture R5. Each non-reference picture is encapsulated into an STAP-B.

Consequently, the transmission order of NAL units is the following:

```
R1, slice group 0, DON 1, carried in MTAP,RTP SN: N
R3, slice group 1, DON 2, carried in MTAP,RTP SN: N
R5, slice group 2, DON 4, carried in MTAP,RTP SN: N
R1, slice group 1, DON 1, carried in MTAP,RTP SN: N+1
R3, slice group 2, DON 2, carried in MTAP,RTP SN: N+1
R5, slice group 0, DON 4, carried in MTAP,RTP SN: N+1
R1, slice group 2, DON 1, carried in MTAP,RTP SN: N+2
R3, slice group 1, DON 2, carried in MTAP,RTP SN: N+2
R5, slice group 0, DON 4, carried in MTAP,RTP SN: N+2
N2, DON 3, carried in STAP-B, RTP SN: N+3
N4, DON 5, carried in STAP-B, RTP SN: N+4
```

The receiver is able to organize the NAL units back in decoding order based on the value of DON associated with each NAL unit.

If one of the MTAPs is lost, the spatially adjacent and temporally co-located macroblocks are received and can be used to conceal the loss efficiently. If one of the STAPs is lost, the effect of the loss does not propagate temporally.

13.3. Example of Robust Packet Scheduling

An example of robust packet scheduling follows. The communication system used in the example consists of the following components in the order that the video is processed from source to sink:

- camera and capturing
- pre-encoding buffer
- encoder
- encoded picture buffer
- transmitter
- transmission channel
- receiver
- receiver buffer
- decoder
- decoded picture buffer
- display

The video communication system used in this example operates as follows. Note that processing of the video stream happens gradually and at the same time in all components of the system. The source video sequence is shot and captured to a pre-encoding buffer. The pre-encoding buffer can be used to order pictures from sampling order to encoding order or to analyze multiple uncompressed frames for bitrate control purposes, for example. In some cases, the pre-encoding buffer may not exist; instead, the sampled pictures are

encoded right away. The encoder encodes pictures from the pre-encoding buffer and stores the output (i.e., coded pictures) to the encoded picture buffer. The transmitter encapsulates the coded pictures from the encoded picture buffer to transmission packets and sends them to a receiver through a transmission channel. The receiver stores the received packets to the receiver buffer. The receiver buffering process typically includes buffering for transmission delay jitter. The receiver buffer can also be used to recover correct decoding order of coded data. The decoder reads coded data from the receiver buffer and produces decoded pictures as output into the decoded picture buffer. The decoded picture buffer is used to recover the output (or display) order of pictures. Finally, pictures are displayed.

In the following example figures, I denotes an IDR picture, R denotes a reference picture, N denotes a non-reference picture, and the number after I, R, or N indicates the sampling time relative to the previous IDR picture in decoding order. Values below the sequence of pictures indicate scaled system clock timestamps. The system clock is initialized arbitrarily in this example, and time runs from left to right. Each I, R, and N picture is mapped into the same timeline compared to the previous processing step, if any, assuming that encoding, transmission, and decoding take no time. Thus, events happening at the same time are located in the same column throughout all example figures.

A subset of a sequence of coded pictures is depicted below in sampling order.

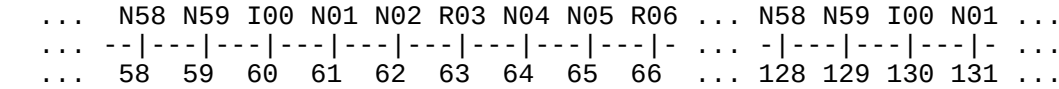


Figure 16. Sequence of pictures in sampling order

The sampled pictures are buffered in the pre-encoding buffer to arrange them in encoding order. In this example, we assume that the non-reference pictures are predicted from both the previous and the next reference picture in output order, except for the non-reference pictures immediately preceding an IDR picture, which are predicted only from the previous reference picture in output order. Thus, the pre-encoding buffer has to contain at least two pictures, and the buffering causes a delay of two picture intervals. The output of the pre-encoding buffering process and the encoding (and decoding) order of the pictures are as follows:

```
... N58 N59 I00 R03 N01 N02 R06 N04 N05 ...
... -|---|---|---|---|---|---|---|---|- ...
... 60  61  62  63  64  65  66  67  68  ...
```

Figure 17. Reordered pictures in the pre-encoding buffer

The encoder or the transmitter can set the value of DON for each picture to a value of DON for the previous picture in decoding order plus one.

For the sake of simplicity, let us assume that:

- the frame rate of the sequence is constant,
- each picture consists of only one slice,
- each slice is encapsulated in a single NAL unit packet,
- there is no transmission delay, and
- pictures are transmitted at constant intervals (that is, 1 / (frame rate)).

When pictures are transmitted in decoding order, they are received as follows:

```
... N58 N59 I00 R03 N01 N02 R06 N04 N05 ...
... -|---|---|---|---|---|---|---|---|- ...
... 60  61  62  63  64  65  66  67  68  ...
```

Figure 18. Received pictures in decoding order

The OPTIONAL sprop-interleaving-depth media type parameter is set to 0, as the transmission (or reception) order is identical to the decoding order.

Initially, the decoder has to buffer for one picture interval in its decoded picture buffer to organize pictures from decoding order to output order, as depicted below:

```
... N58 N59 I00 N01 N02 R03 N04 N05 R06 ...
... -|---|---|---|---|---|---|---|---|- ...
... 61  62  63  64  65  66  67  68  69  ...
```

Figure 19. Output order

The amount of required initial buffering in the decoded picture buffer can be signaled in the buffering period SEI message or with the num_reorder_frames syntax element of H.264 video usability information. num_reorder_frames indicates the maximum number of frames, complementary field pairs, or non-paired fields that precede any frame, complementary field pair, or non-paired field in the

sequence in decoding order and that follow it in output order. For the sake of simplicity, we assume that num_reorder_frames is used to indicate the initial buffer in the decoded picture buffer. In this example, num_reorder_frames is equal to 1.

It can be observed that if the IDR picture I00 is lost during transmission and a retransmission request is issued when the value of the system clock is 62, there is one picture interval of time (until the system clock reaches timestamp 63) to receive the retransmitted IDR picture I00.

Let us then assume that IDR pictures are transmitted two frame intervals earlier than their decoding position; that is, the pictures are transmitted as follows:

```
...  I00 N58 N59 R03 N01 N02 R06 N04 N05 ...
... --|---|---|---|---|---|---|---|---|- ...
...  62  63  64  65  66  67  68  69  70  ...
```

Figure 20. Interleaving: Early IDR pictures in sending order

The OPTIONAL sprop-interleaving-depth media type parameter is set equal to 1 according to its definition. (The value of sprop-interleaving-depth in this example can be derived as follows: picture I00 is the only picture preceding picture N58 or N59 in transmission order and following it in decoding order. Except for pictures I00, N58, and N59, the transmission order is the same as the decoding order of pictures. As a coded picture is encapsulated into exactly one NAL unit, the value of sprop-interleaving-depth is equal to the maximum number of pictures preceding any picture in transmission order and following the picture in decoding order).

The receiver buffering process contains two pictures at a time according to the value of the sprop-interleaving-depth parameter and orders pictures from the reception order to the correct decoding order based on the value of DON associated with each picture. The output of the receiver buffering process is as follows:

```
... N58 N59 I00 R03 N01 N02 R06 N04 N05 ...
... -|---|---|---|---|---|---|---|---|- ...
... 63  64  65  66  67  68  69  70  71  ...
```

Figure 21. Interleaving: Receiver buffer

Again, an initial buffering delay of one picture interval is needed to organize pictures from decoding order to output order, as depicted below:

```
... N58 N59 I00 N01 N02 R03 N04 N05 ...
... -|---|---|---|---|---|---|---|- ...
... 64  65  66  67  68  69  70  71  ...
```

Figure 22. Interleaving: Receiver buffer after reordering

Note that the maximum delay that IDR pictures can undergo during transmission, including possible application, transport, or link layer retransmission, is equal to three picture intervals. Thus, the loss resiliency of IDR pictures is improved in systems supporting retransmission compared to the case in which pictures are transmitted in their decoding order.

13.4. Robust Transmission Scheduling of Redundant Coded Slices

A redundant coded picture is a coded representation of a picture or a part of a picture that is not used in the decoding process if the corresponding primary coded picture is correctly decoded. There should be no noticeable difference between any area of the decoded primary picture and a corresponding area that would result from application of the H.264 decoding process for any redundant picture in the same access unit. A redundant coded slice is a coded slice that is a part of a redundant coded picture.

Redundant coded pictures can be used to provide unequal error protection in error-prone video transmission. If a primary coded representation of a picture is decoded incorrectly, a corresponding redundant coded picture can be decoded. Examples of applications and coding techniques using the redundant codec picture feature include the video redundancy coding [23] and the protection of "key pictures" in multicast streaming [24].

One property of many error-prone video communications systems is that transmission errors are often bursty. Therefore, they may affect more than one consecutive transmission packet in transmission order. In low bitrate video communication, it is relatively common for an entire coded picture to be encapsulated into one transmission packet. Consequently, a primary coded picture and the corresponding redundant coded pictures may be transmitted in consecutive packets in transmission order. To make the transmission scheme more tolerant of bursty transmission errors, it is beneficial to transmit the primary coded picture and redundant coded picture separated by more than a single packet. The DON concept enables this.

13.5. Remarks on Other Design Possibilities

The slice header syntax structure of the H.264 coding standard contains the frame_num syntax element that can indicate the decoding order of coded frames. However, the usage of the frame_num syntax element is not feasible or desirable to recover the decoding order, due to the following reasons:

- The receiver is required to parse at least one slice header per coded picture (before passing the coded data to the decoder).
- Coded slices from multiple coded video sequences cannot be interleaved, as the frame number syntax element is reset to 0 in each IDR picture.
- The coded fields of a complementary field pair share the same value of the frame_num syntax element. Thus, the decoding order of the coded fields of a complementary field pair cannot be recovered based on the frame_num syntax element or any other syntax element of the H.264 coding syntax.

The RTP payload format for transport of MPEG-4 elementary streams [25] enables interleaving of access units and transmission of multiple access units in the same RTP packet. An access unit is specified in the H.264 coding standard to comprise all NAL units associated with a primary coded picture according to Subclause 7.4.1.2 of [1]. Consequently, slices of different pictures cannot be interleaved, and the multi-picture slice interleaving technique (see Section 12.6) for improved error resilience cannot be used.

14. Changes from RFC 3984

Following is the list of technical changes (including bug fixes) from RFC 3984. Besides this list of technical changes, numerous editorial changes have been made, but not documented in this section. Note that Section 8.2.2 is where much of the important changes in this memo occurs and deserves particular attention.

1) In Sections 5.4, 5.5, 6.2, 6.3, and 6.4, removed that the packetization mode in use may be signaled by external means.

2) In Section 7.2.2, changed the sentence

   There are N VCL NAL units in the de-interleaving buffer.

   to

   There are N or more VCL NAL units in the de-interleaving buffer.

3) In Section 8.1, the semantics of sprop-init-buf-time (paragraph 2), changed the sentence

   The parameter is the maximum value of (transmission time of a NAL unit - decoding time of the NAL unit), assuming reliable and instantaneous transmission, the same timeline for transmission and decoding, and that decoding starts when the first packet arrives.

   to

   The parameter is the maximum value of (decoding time of the NAL unit - transmission time of a NAL unit), assuming reliable and instantaneous transmission, the same timeline for transmission and decoding, and that decoding starts when the first packet arrives.

4) Added media type parameters max-smbps, sprop-level-parameter-sets, use-level-src-parameter-sets, in-band-parameter-sets, sar-understood, and sar-supported.

5) In Section 8.1, removed the specification of parameter-add. Other descriptions of parameter-add (in Sections 8.2 and 8.4) were also removed.

6) In Section 8.1, added a constraint to sprop-parameter-sets such that it can only contain parameter sets for the same profile and level as indicated by profile-level-id.

7) In Section 8.2.1, added that sprop-parameter-sets and sprop-level-parameter-sets may be either included in the "a=fmtp" line of SDP or conveyed using the "fmtp" source attribute as specified in Section 6.3 of [9].

8) In Section 8.2.2, removed sprop-deint-buf-req from being part of the media format configuration in usage with the SDP Offer/Answer model.

9) In Section 8.2.2, made it clear that level is downgradable in the SDP Offer/Answer model, i.e., the use of the level part of profile-level-id does not need to be symmetric (the level included in the answer can be lower than or equal to the level included in the offer).

10) In Section 8.2.2, removed that the capability parameters may be used to declare encoding capabilities.

11) In Section 8.2.2, added rules on how to use sprop-parameter-sets and sprop-level-parameter-sets for out-of-band transport of parameter sets, with or without level downgrading.

12) In Section 8.2.2, clarified the rules of using the media type parameters with SDP Offer/Answer for multicast.

13) In Section 8.2.2, completed and corrected the list of how different media type parameters shall be interpreted in the different combinations of offer or answer and direction attribute.

14) In Section 8.4, changed the text such that both out-of-band and in-band transport of parameter sets are allowed, and neither is recommended or required.

15) Added Section 8.5 (informative) providing example methods for decoder refresh to handle parameter set losses.

16) Added media type parameters max-recv-level and level-asymmetry-allowed and adjusted associated text and examples for level upgrade and asymmetry.

15. Backward Compatibility to RFC 3984

The current document is a revision of RFC 3984 and obsoletes it. The technical changes relative to RFC 3984 are listed in Section 14. This section addresses the backward compatibility issues.

It should be noted that for the majority of cases, there will be no compatibility issues for legacy implementations per RFC 3984 and new implementations per this document to interwork. Compatibility issues may only occur when both of the following conditions are true: 1) legacy implementations and new implementations are interworking, and 2) parameter sets are transported out-of-band. When such compatibility issues occur, it is easy to debug and find the reason for the incompatibility using the following analyses.

Items 1, 2, 3, 7, 9, 10, 12, and 13 are bug-fix types of changes and do not incur any backward compatibility issues.

Item 4 (addition of six new media type parameters) does not incur any backward compatibility issues for SDP Offer/Answer-based applications, as legacy RFC 3984 receivers ignore these parameters, and it is fine for legacy RFC 3984 senders not to use these parameters as they are optional. However, there is a backward compatibility issue for declarative-usage-based applications (only for the parameter sprop-level-parameter-sets as the other five

parameters are not usable in declarative usage). For example, declarative-usage-based applications using RTSP and SAP have a backward compatibility issue because the SDP receiver per RFC 3984 cannot accept a session for which the SDP includes an unrecognized parameter. Therefore, the RTSP or SAP server may have to prepare two sets of streams, one for legacy RFC 3984 receivers and one for receivers according to this memo.

Items 5, 6, and 11 are related to out-of-band transport of parameter sets. There are following backward compatibility issues.

1) When a legacy sender per RFC 3984 includes parameter sets for a level different than the default level indicated by profile-level-id to sprop-parameter-sets, the parameter value of sprop-parameter-sets is invalid to the receiver per this memo; therefore, the session may be rejected.

2) In SDP Offer/Answer between a legacy offerer per RFC 3984 and an answerer per this memo, when the answerer includes in the answer parameter sets that are not a superset of the parameter sets included in the offer, the parameter value of sprop-parameter-sets is invalid to the offerer, and the session may not be initiated properly (related to change item 11).

3) When one endpoint A per this memo includes in-band-parameter-sets equal to 1, the other side B per RFC 3984 does not understand that it must transmit parameter sets in-band, and B may still exclude parameter sets in the in-band stream it is sending. Consequently, endpoint A cannot decode the stream it receives.

Item 7 (allowance of conveying sprop-parameter-sets and sprop-level-parameter-sets using the "fmtp" source attribute as specified in Section 6.3 of [9]) is similar to item 4. It does not incur any backward compatibility issues for SDP Offer/Answer-based applications, as legacy RFC 3984 receivers ignore the "fmtp" source attribute, and it is fine for legacy RFC 3984 senders not to use the "fmtp" source attribute as it is optional. However, there is a backward compatibility issue for SDP declarative-usage-based applications, e.g., those using RTSP and SAP, because the SDP receiver per RFC 3984 cannot accept a session for which the SDP includes an unrecognized parameter (i.e., the "fmtp" source attribute). Therefore, the RTSP or SAP server may have to prepare two sets of streams, one for legacy RFC 3984 receivers and one for receivers according to this memo.

Item 14 does not incur any backward compatibility issues, as out-of-band transport of parameter sets is still allowed.

Item 15 does not incur any backward compatibility issues, as the added Section 8.5 is informative.

Item 16 does not create any backward compatibility issues as the handling of the default level is the same if either end is RFC 3984 compliant, and, furthermore, RFC-3984-compliant ends would simply ignore the new media type parameters, if present.

16. Acknowledgements

Stephan Wenger, Miska Hannuksela, Thomas Stockhammer, Magnus Westerlund, and David Singer are thanked as the authors of RFC 3984. Dave Lindbergh, Philippe Gentric, Gonzalo Camarillo, Gary Sullivan, Joerg Ott, and Colin Perkins are thanked for careful review during the development of RFC 3984. Stephen Botzko, Magnus Westerlund, Alex Eleftheriadis, Thomas Schierl, Tom Taylor, Ali Begen, Aaron Wells, Stuart Taylor, Robert Sparks, Dan Romascanu, and Niclas Comstedt are thanked for their valuable comments and input during the development of this memo.

17. References

17.1. Normative References

[1] ITU-T Recommendation H.264, "Advanced video coding for generic audiovisual services", March 2010.

[2] ISO/IEC International Standard 14496-10:2008.

[3] ITU-T Recommendation H.241, "Extended video procedures and control signals for H.300-series terminals", May 2006.

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

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

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

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

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

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

17.2. Informative References

[10] Luthra, A., Sullivan, G.J., and T. Wiegand (eds.), "Introduction to the special issue on the H.264/AVC video coding standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, July 2003.

[11] Ott, J., Bormann, C., Sullivan, G., Wenger, S., and R. Even, Ed., "RTP Payload Format for ITU-T Rec. H.263 Video", RFC 4629, January 2007.

[12] ISO/IEC International Standard 14496-2:2004.

[13] Wenger, S., "H.264/AVC over IP", IEEE Transaction on Circuits and Systems for Video Technology, Vol. 13, No. 7, July 2003.

[14] Wenger, S., "H.26L over IP: The IP-Network Adaptation Layer", Proceedings Packet Video Workshop, April 2002.

[15] Stockhammer, T., Hannuksela, M.M., and S. Wenger, "H.26L/JVT Coding Network Abstraction Layer and IP-Based Transport", IEEE International Conference on Image Processing (ICIP 2002), Rochester, NY, September 2002.

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

[17] ITU-T Recommendation H.223, "Multiplexing protocol for low bit rate multimedia communication", July 2001.

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

[19] Stockhammer, T., Wiegand, T., Oelbaum, T., and F. Obermeier, "Video Coding and Transport Layer Techniques for H.264/AVC-Based Transmission over Packet-Lossy Networks", IEEE International Conference on Image Processing (ICIP 2003), Barcelona, Spain, September 2003.

[20] Varsa, V. and M. Karczewicz, "Slice interleaving in compressed video packetization", Packet Video Workshop 2000.

[21] Kang, S.H. and A. Zakhor, "Packet scheduling algorithm for wireless video streaming", Packet Video Workshop 2002.

[22] Hannuksela, M.M., "Enhanced Concept of GOP", JVT-B042, available http://ftp3.itu.int/av-arch/video-site/0201_Gen/JVT-B042.doc, January 2002.

[23] Wenger, S., "Video Redundancy Coding in H.263+", 1997 International Workshop on Audio-Visual Services over Packet Networks, September 1997.

[24] Wang, Y.-K., Hannuksela, M.M., and M. Gabbouj, "Error Resilient Video Coding Using Unequally Protected Key Pictures", in Proc. International Workshop VLBV03, September 2003.

[25] van der Meer, J., Mackie, D., Swaminathan, V., Singer, D., and P. Gentric, "RTP Payload Format for Transport of MPEG-4 Elementary Streams", RFC 3640, November 2003.

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

[27] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.

[28] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, October 2000.

[29] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 5117, January 2008.

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

Authors' Addresses

Ye-Kui Wang
Huawei Technologies
400 Crossing Blvd, 2nd Floor
Bridgewater, NJ 08807
USA

Phone: +1-908-541-3518
EMail: yekui.wang@huawei.com

Roni Even
Huawei Technologies
14 David Hamelech
Tel Aviv 64953
Israel

Phone: +972-545481099
EMail: even.roni@huawei.com

Tom Kristensen
TANDBERG
Philip Pedersens vei 22
N-1366 Lysaker
Norway

Phone: +47 67125125
EMail: tom.kristensen@tandberg.com, tomkri@ifi.uio.no

Randell Jesup
WorldGate Communications
3800 Horizon Blvd, Suite #103
Trevose, PA 19053-4947
USA

Phone: +1-215-354-5166
EMail: rjesup@wgate.com, randell_ietf@jesup.org