Network Working Group
Internet Draft
Document: draft-ietf-avt-rtp-h264-11.txt
Expires: February 2005

S. Wenger
M.M. Hannuksela
T. Stockhammer
M. Westerlund
D. Singer

August 2004

RTP payload Format for H.264 Video

Status of this Memo

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. 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

supporting from simple low-bit rate conversational usage to Internet video streaming with interleaved transmission, all the way to high bit-rate video-on-demand applications.

Table of Contents

1. Introduction

1.1. The H.264 codec

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 Part 10 (both also known as Advanced Video Coding, AVC) [2]. Recommendation H.264 was approved by ITU-T on May 2003, and the approved draft specification is available for public review [9]. In this memo the H.264 acronym is used for the codec and the standard, but the memo is equally applicable to the ISO/IEC counterpart of the coding standard.

The H.264 video codec has a very broad application range that covers all forms of digital compressed video from low bit rate Internet streaming applications to HDTV broadcast and Digital Cinema applications with near loss-less coding. The overall performance of H.264 is as such that bit rate savings of 50% or more, compared to the current state of technology, 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 distinguishes conceptually 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, motion compensated prediction, and a loop filter. It follows the general concept of most of today's video codecs, a macroblock-based coder that utilizes 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).

Macroblocks in slices are ordered in scan order unless a different macroblock allocation is specified, using the so-called Flexible

Macroblock Ordering syntax. In-picture prediction is used only within a slice. More information is provided in [9].

The Network Abstraction Layer (NAL) encoder encapsulates the slice output of the VCL encoder into Network Abstraction Layer Units (NAL units), which are suitable for the transmission over packet networks or the use in packet oriented multiplex environments. Annex B of H.264 defines an encapsulation process to transmit such NAL units over byte-stream 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 common in the form of the Temporal Reference in earlier video compression standards). Also, there are NAL units that affect many pictures and are, hence, inherently time-less. For this reason, the handling of the RTP timestamp requires some special considerations for those NAL units for which the sampling or presentation time is not defined, or, at transmission time, 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 2429 [12] or MPEG-4's Header Extension Code (HEC) [13] unnecessary. The way that this was achieved is to decouple information that is 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 such 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 set and picture parameter set. 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.

In order to be able to change picture parameters (such as the picture size), without having the need 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.

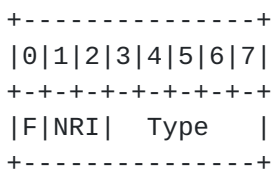
This mechanism allows the decoupling of the transmission of parameter sets from the packet stream, and the transmission of them by external means, e.g. as a side effect of the capability exchange, or through a (reliable or unreliable) control protocol. It may even be possible that they get never transmitted 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 [14], [15] and [16].

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

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 [3].

This specification uses the notion of setting and clearing a bit when handling bit fields. 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 below 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 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].

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 on 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

signalling (e.g. to learn about the payload type mappings of

the media streams) and that it has to be trusted when working with SRTP. The advantage of using MANEs is that they allow to drop packets according to the needs of the media coding. For example, if a MANE needs to drop packets due to congestion on a certain link, it can identify those packets whose dropping has the smallest negative impact on the user experience, and remove those in order to remove the congestion and/or keep the delay low.

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
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

5. RTP Payload Format

5.1. RTP Header Usage

The format of the RTP header is specified in RFC 3550 [4] and reprinted in Figure 1 for convenience. This payload format uses the fields of the header in a manner consistent with that specification.

When encapsulating one NAL unit 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 and 5.8, respectively.

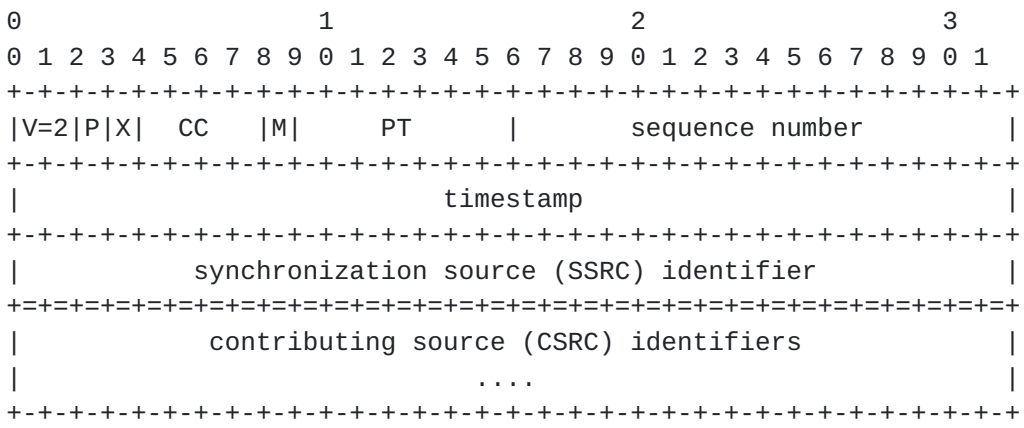


Figure 1: RTP header according RFC 3550.

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 and 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 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, and 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 needs 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.

RTP senders SHOULD NOT transmit picture timing SEI messages for pictures that are not supposed to be displayed as multiple fields.

In case that 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 SEI message picture timing values. 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 where film content that consists of coded frames is displayed on an 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.
>
> Informative note: Due to the fact that H.264 allows the decoding order to be different from the display order, values of RTP timestamps may not be monotonically non-decreasing as a function of RTP sequence numbers. Furthermore, the value for interarrival jitter reported in the RTCP reports may not be a trustworthy indication of the network performance, as the calculation rules for interarrival jitter (section 6.4.1

of RFC 3550) assume that the RTP timestamp of a packet is directly proportional to its transmission time.

5.2. Common structure of the RTP payload format

The payload format defines three different basic payload structures. A receiver can identify the payload structure by the first byte of the RTP 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:

Single NAL Unit Packet: Contains only a single NAL unit in the payload. The NAL header type field will be 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.

Table 1. Summary of NAL unit types and their payload structures.

Type	Packet	Type name	Section
0	undefined		-
1-23	NAL unit	Single NAL unit packet per H.264	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	undefined		-

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.

5.3. NAL Unit Octet Usage

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

```
+---------------+
|0|1|2|3|4|5|6|7|
+-+-+-+-+-+-+-+-+
|F|NRI|  Type   |
+---------------+
```

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 is equal to 0. When the F bit is set, the decoder is advised that bit errors or any other syntax violation may be present in the payload or in the NAL unit type octet. The simplest decoder reaction to respond 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 compared to 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 greater than 00 indicate the relative transport priority, as determined by the encoder. MANEs can use this information to protect more important NAL units better than less important NAL units. 11 is the highest transport priority, followed by 10, then by 01 and, 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.

An H.264 encoder SHOULD set the value of NRI for NAL units having nal_unit_type equal to 7 or 8 (indicating a sequence parameter set or a picture parameter set respectively) to 11 (in binary format). An H.264 encoder SHOULD set the value of NRI 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) to 11 (in binary format).

The following example for a mapping of the remaining nal_unit_types to NRI values MAY be used and has been shown as efficient in a certain environment [15]. Other mappings MAY also be desirable, depending on the application and the H.264/AVC Annex A 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 bit stream 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/AVC.

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, because 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 [17] (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 MIME parameter or by external means. The used packetization mode governs which NAL unit types are allowed in RTP payloads. Table 3 summarizes the allowed NAL unit types for each packetization mode. Some NAL unit type values (indicated as undefined in Table 3) are reserved for future extensions. NAL units of those types SHOULD NOT be sent by a sender, and MUST be ignored by a receiver. For example, the 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". 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).

Type	Packet	Single NAL Unit Mode	Non-Interleaved Mode	Interleaved Mode
0	undefined	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	undefined	ig	ig	ig

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 example 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 MIME parameter as follows. When the value of the OPTIONAL sprop-interleaving-depth MIME parameter is equal to 0 (explicitly or per default) or transmission of NAL units out of their decoding order is disallowed by external means, the transmission order of NAL units MUST conform to the NAL unit decoding order. When the value of the OPTIONAL sprop-interleaving-depth MIME parameter is greater than 0 or transmission of NAL units out of their decoding order is allowed by external means,

- 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 decapsulating 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 and 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 a STAP, and second provided through the RTP sequence number for the order between STAPs, FUs, and single NAL unit packets.

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, then 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 NAL unit in decoding order incremented by one.

An example decapsulation 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 is equal to one even in case of error-free transmission. An increment by one is not required, because 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. At the time of transmitting the first intra picture, the originator does not exactly know how many NAL units are going to 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 at the time of transmitting them and gaps in values of DON may occur.

5.6. Single NAL Unit Packet

The single NAL unit packet defined here MUST contain one and 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 decapsulating 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.

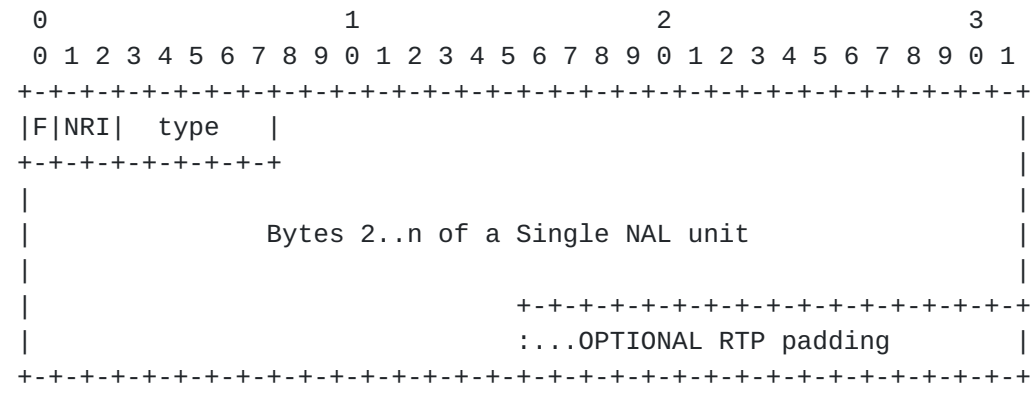


Figure 2. RTP payload format for single NAL unit packet.

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 or non-IP (e.g. ITU-T H.324/M) based wireless communication systems with preferred transmission unit sizes of 254 bytes or less. In order 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-time. Two types of STAPs are defined, one without DON (STAP-A) and another one including DON (STAP-B).
- Multi-time aggregation packet (MTAP) aggregates NAL units with potentially differing NALU-time. Two different MTAPs are defined that differ in the length of the NAL unit timestamp offset.

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

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

The structure of the RTP payload format for aggregation packets is

presented in Figure 3.

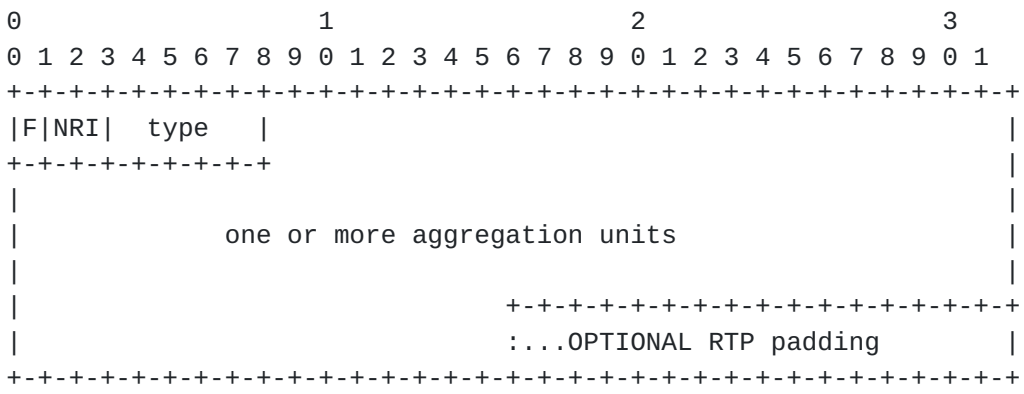


Figure 3. RTP payload format for aggregation packets.

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 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 section 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 such that the resulting IP packet is smaller than the MTU size. An aggregation packet MUST NOT contain fragmentation units specified in section 5.8. Aggregation packets MUST NOT be nested, i.e., an aggregation packet MUST NOT contain another aggregation packet.

5.7.1. Single-Time Aggregation Packet

Single-time aggregation packet (STAP) SHOULD be used whenever aggregating NAL units 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.

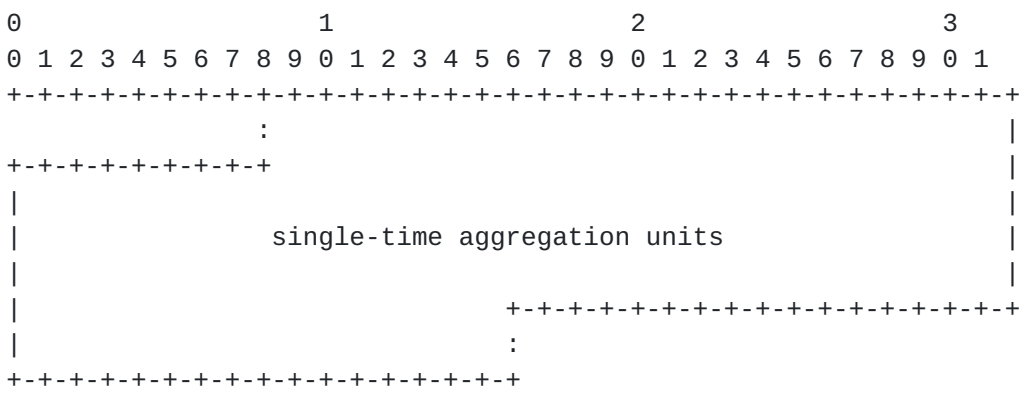


Figure 4. Payload format for STAP-A.

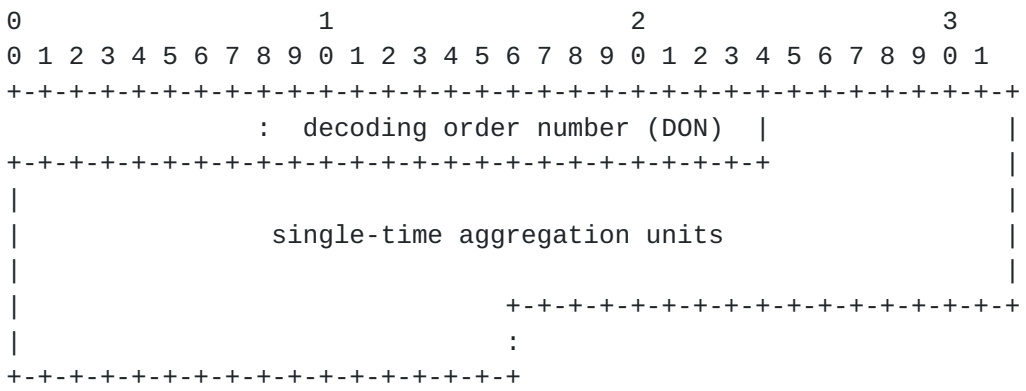


Figure 5. Payload format for STAP-B.

The DON field specifies the value of DON for the first NAL unit in an STAP-B in transmission order. The value of DON for each successive NAL unit in appearance order in an STAP-B 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.

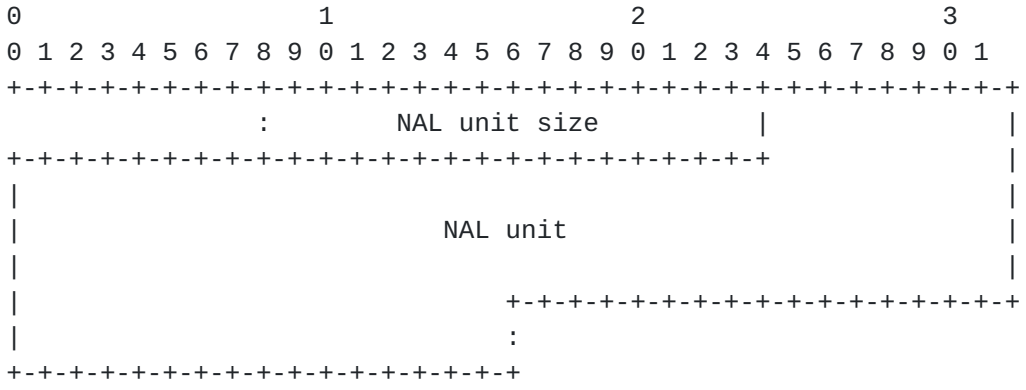


Figure 6. Structure for single-time aggregation unit.

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.

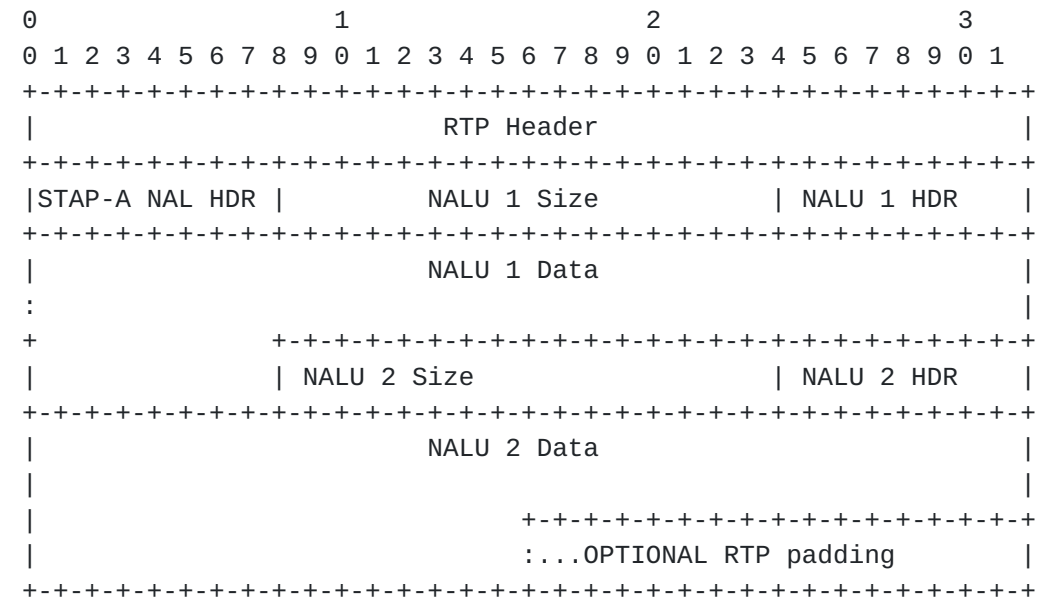


Figure 7. An example of an RTP packet including a STAP-A and two single-time aggregation units.

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.

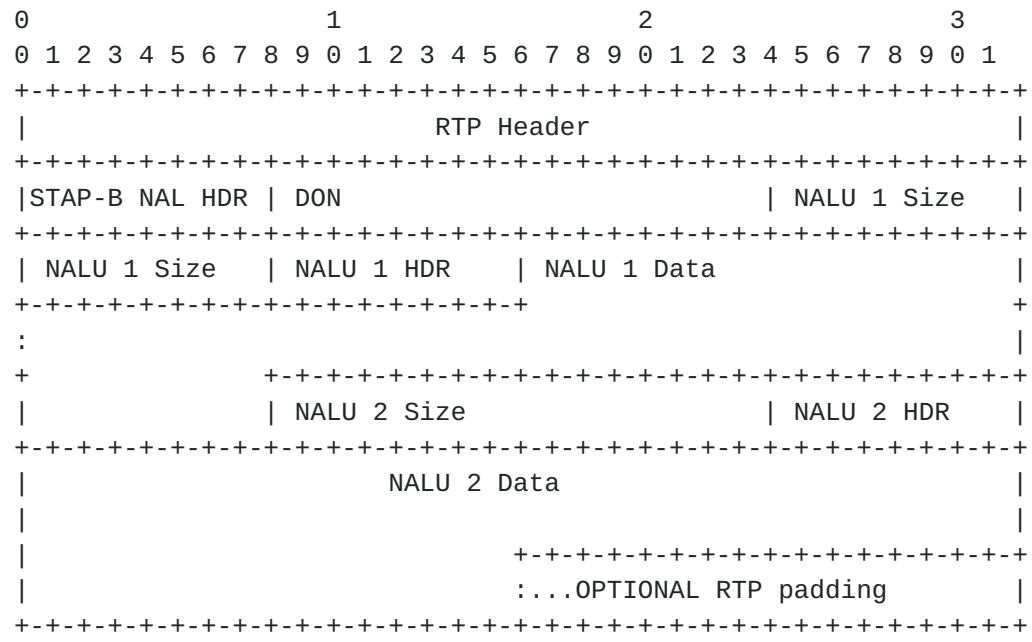


Figure 8. An example of an RTP packet including an STAP-B and two single-time aggregation units.

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 the NAL units are encapsulated in an MTAP.

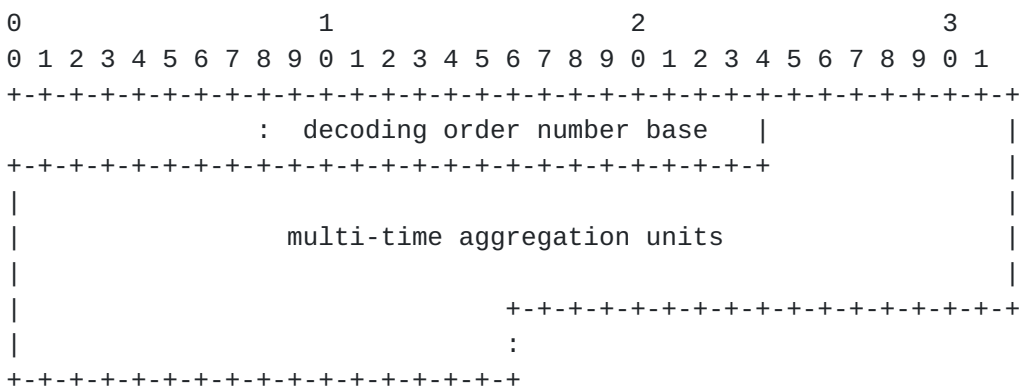


Figure 9. NAL unit payload format for MTAPs.

Two different multi-time aggregation units are defined in this specification. Both of them consist of 16 bits 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 is the flexibility of the MTAP, but the higher is also the overhead.

The structure of the multi-time aggregation units for MTAP16 and MTAP24 are presented in Figure 10 and Figure 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. DON of the following NAL 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).

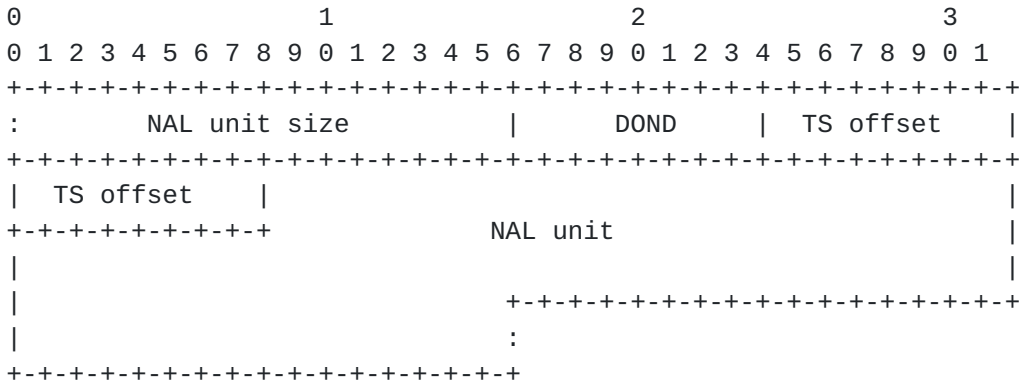


Figure 10. Multi-time aggregation unit for MTAP16

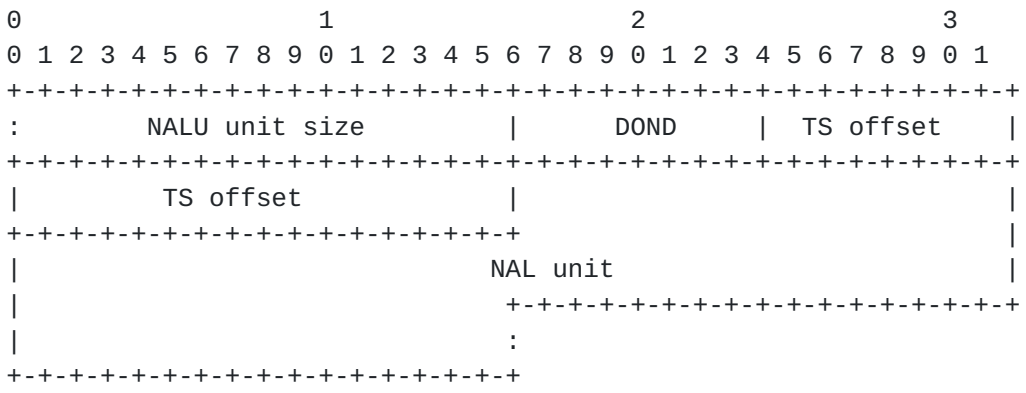


Figure 11. Multi-time aggregation unit for MTAP24

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 has the smallest extended RTP timestamp among all the aggregation units of an MTAP if 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 wrap around of the timestamp field, thus enabling one to actually determine the smallest value if the timestamp wraps. Such an "earliest" aggregation unit may not be the first one in the order 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.

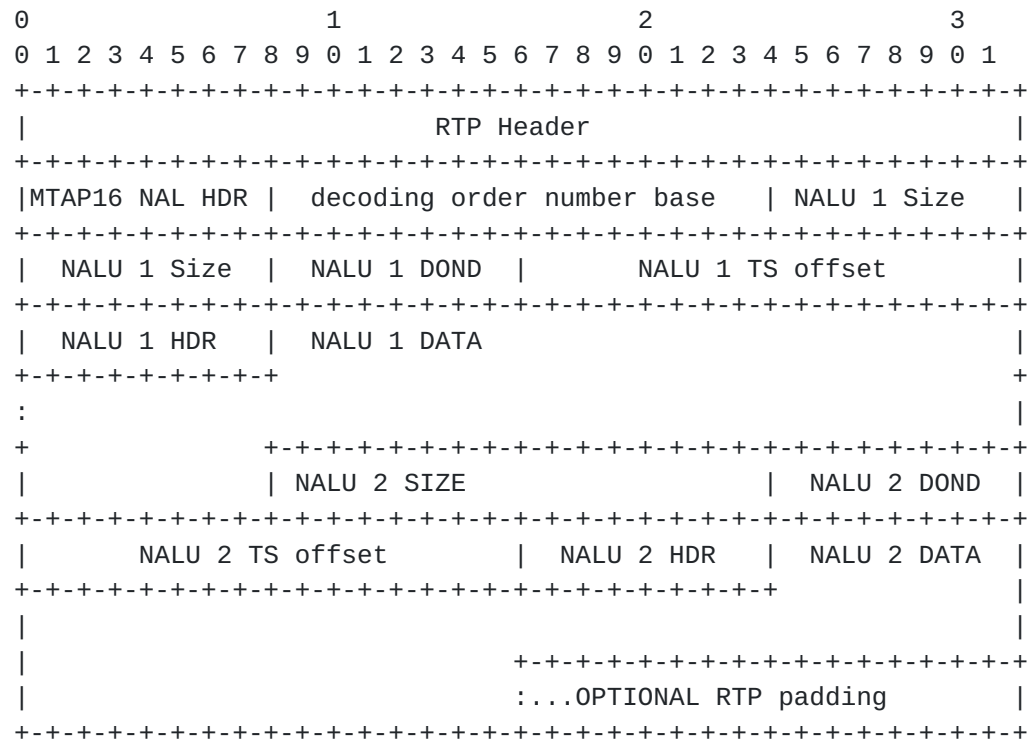


Figure 12. An example of an RTP packet including a multi-time aggregation packet of type MTAP16 and two multi-time aggregation units.

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.

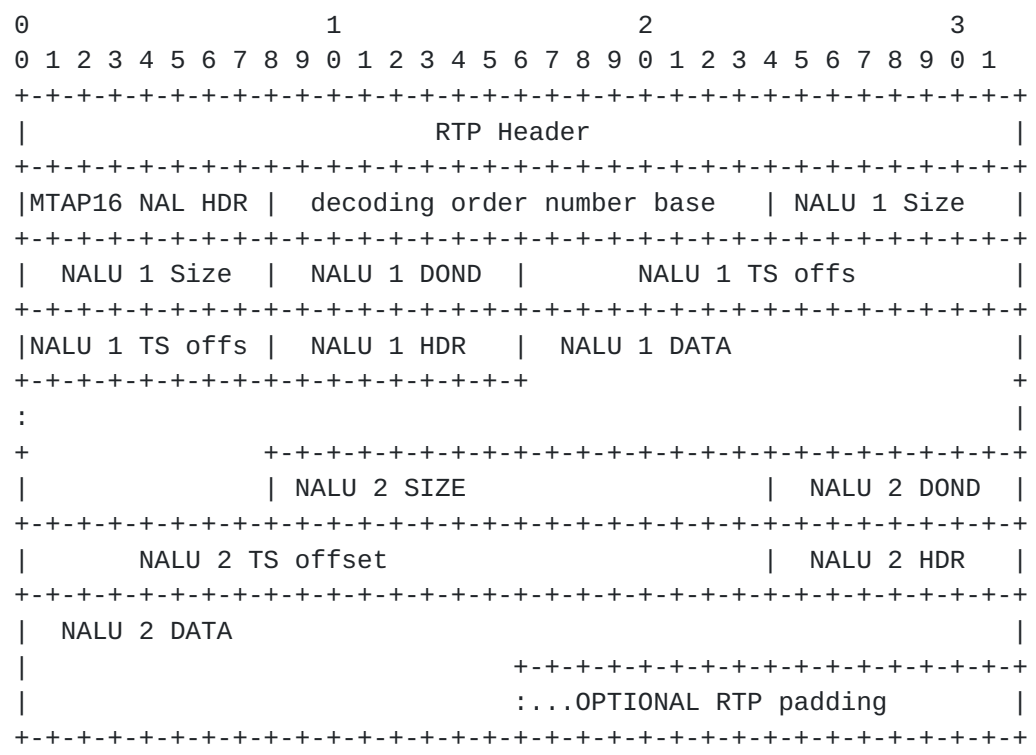


Figure 13. An example of an RTP packet including a multi-time aggregation packet of type MTAP16 and two multi-time aggregation units.

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 picture 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 fragmented NAL unit. STAPs and MTAPs MUST NOT be fragmented. FUs MUST NOT be nested, i.e., 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.

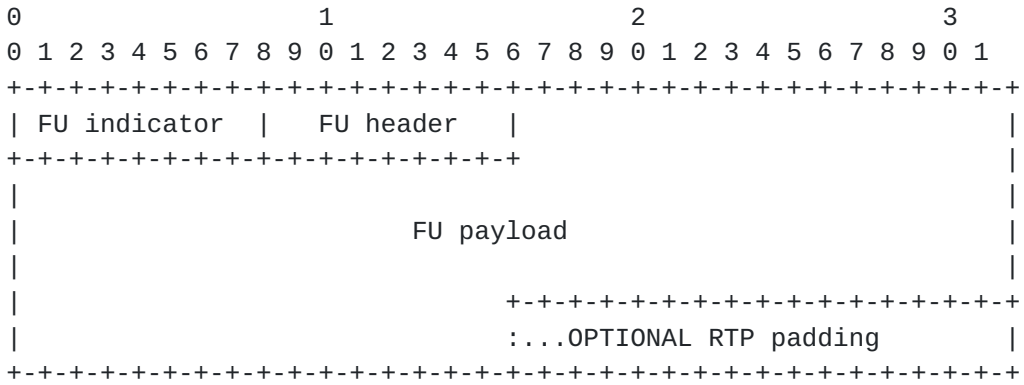


Figure 14. RTP payload format for FU-A.

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.

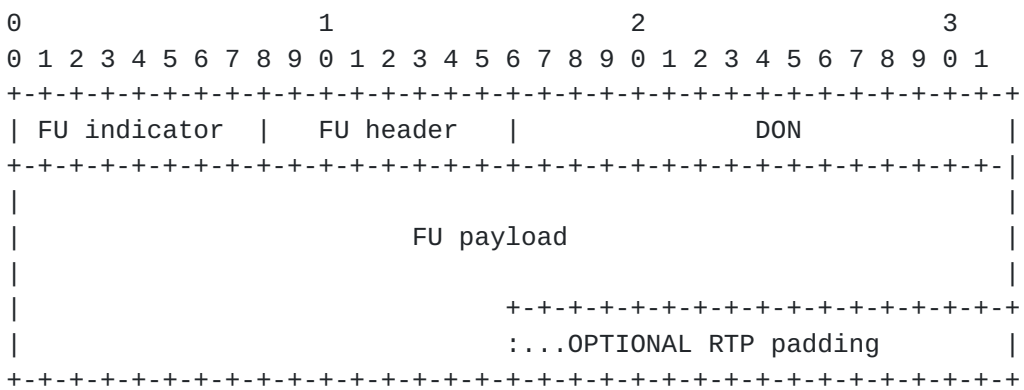
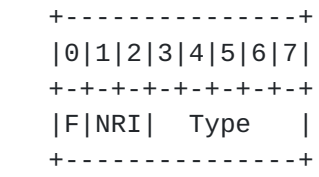


Figure 15. RTP payload format for FU-B.

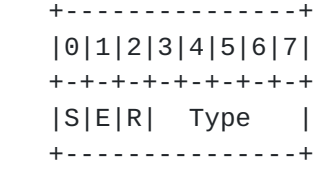
NAL unit type FU-B MUST be used in the interleaved packetization mode for the first fragmentation unit of a fragmented NAL unit. NAL unit type FU-B MUST NOT be used in any other case. In other words, in the interleaved packetization mode, each NALU that is fragmented has an FU-B as the first fragment, 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
The Start bit, when one, indicates the start of a fragmented NAL unit. Otherwise, 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
The End bit, when one, 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. Otherwise, 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, i.e., 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 such that if the fragmentation unit payloads of consecutive FUs are sequentially concatenated, the payload of the fragmented NAL unit is 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 F and NRI fields of the FU indicator octet of the fragmentation unit and in the type field of the FU header. A 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 near loss-less environments. Those senders can be characterized in that they packetize NALU fragments before the NALU is completely generated and hence, before the NALU size if 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 NALUs should be carefully weighted against the increase of the risk of the loss of the NALU, because of the additional packets that are 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 that are 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 hence sharing the same RTP timestamp value) MAY be sent in any order permitted by the applicable profile defined in [1], although, for delay-critical systems, they SHOULD be sent in their original coding order to minimize the delay. Note that the coding order is not necessarily the scan order, but the order the NAL packets become available to the RTP stack.

- 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, because 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 according to the non-interleaved mode and the interleaved mode MUST enforce the following packetization rule:

- 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, in an RTP translator. 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 2733 [21], 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 RTCP as per RFC 3550.

6.2. Single NAL Unit Mode

This mode is in use when the value of the OPTIONAL packetization-mode MIME parameter is equal to 0 or packetization-mode is not

present or no other packetization mode is signaled by external

means. 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 [17] (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 MIME parameter is equal to 1 or the mode is turned on by external means. 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 MIME parameter is equal to 2 or the mode is turned on by external means. 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 (Informative)

The de-packetization process is implementation dependent. Hence, the following description should be seen as an example of a suitable implementation. Other schemes may be used as well. 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 decapsulation 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 sequences number and the RTP timestamp) are removed. To determine the exact time for decoding, factors such as a possible intentional delay to allow for proper inter-stream synchronization must be factored in.

7.1. Single NAL Unit and Non-Interleaved Mode

The receiver includes a receiver buffer to compensate transmission delay jitter. The receiver stores incoming packets in reception order into the receiver buffer. Packets are decapsulated in RTP sequence number order. If a decapsulated packet is a single NAL unit packet, the NAL unit contained in the packet is passed directly to the decoder. If a decapsulated packet is an STAP-A, the NAL units contained in the packet are passed to the decoder in the order they are encapsulated in the packet. If a decapsulated packet is an FU-A, all the fragments of the fragmented NAL unit are concatenated and 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 packets from transmission order to the NAL unit decoding order. In this section, the receiver operation is described assuming that there is no transmission delay jitter. To make a difference between a practical receiver buffer that is also used for compensation of transmission delay jitter, the receiver buffer is hereinafter called the deinterleaving buffer in this section. Receivers SHOULD also prepare for transmission delay jitter, i.e., either reserve separate buffers for transmission delay jitter buffering and deinterleaving buffering or use a receiver buffer for both transmission delay jitter and deinterleaving. 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 deinterleaving buffer. Subsection 7.2.2 specifies the receiver process how to organize received NAL units to the NAL unit decoding order.

7.2.1. Size of the Deinterleaving Buffer

When 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 deinterleaving buffer with the deint-buf-cap MIME parameter. The sender indicates the requirement for the deinterleaving buffer size with the sprop-deint-buf-req MIME parameter. It is therefore RECOMMENDED to set the deinterleaving buffer size, in terms of number of bytes, equal to or greater than the value of sprop-deint-buf-req MIME parameter. See section 8.1 for further information on deint-buf-cap and sprop-deint-buf-req MIME parameters and section 8.2.2 for further information on their use in SDP Offer/Answer model.

When a declarative session description is used in session setup, the sprop-deint-buf-req MIME parameter signals the requirement for the deinterleaving buffer size. It is therefore RECOMMENDED to set the deinterleaving buffer size, in terms of number of bytes, equal to or greater than the value of sprop-deint-buf-req MIME parameter.

7.2.2. Deinterleaving 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 is started and the buffering-while-playing mode is used.

Regardless of the buffering state the receiver stores incoming NAL units in reception order into the deinterleaving buffer as follows. NAL units of aggregation packets are stored into the deinterleaving buffer individually. The value of DON is calculated and stored for all NAL units.

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 MIME type parameter (see section 8.1) incremented by 1.

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

- There are N VCL NAL units in the deinterleaving 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 MIME parameter.

The NAL units to be removed from the deinterleaving buffer are

determined as follows:

- o If the deinterleaving buffer contains at least N VCL NAL units, NAL units are removed from the deinterleaving buffer and passed to the decoder in the order specified below until the buffer contains N-1 VCL NAL units.
- o 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 deinterleaving 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 received NAL units.
- o

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

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

- o Intelligent RTP receivers (e.g. in gateways) may identify lost coded slice data partitions A (DPAs). If a lost DPA is found, 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.

- o 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.

- o 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 need 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 bit stream. The parameters are specified here as part of the MIME subtype registration for the ITU-T H.264 | ISO/IEC 14496-10 codec. A mapping of the parameters into the Session Description Protocol (SDP) [5] is also provided for those applications that use SDP. Equivalent parameters could be defined elsewhere for use with control protocols that do not use MIME or SDP.

Some parameters provide a receiver with the properties of the stream that is going to be sent. The name of all these parameters starts 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. The media sender selects all "sprop" parameters rather than the receiver. This uncommon characteristic of the "sprop" parameters may not be compatible with some signaling protocol concepts, in which case the use of these parameters SHOULD be avoided.

8.1. MIME Registration

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

The receiver MUST ignore any unspecified parameter.

Media Type name: video

Media subtype name: H264

Required parameters: none

OPTIONAL parameters:
profile-level-id: A base16 [6] (hexadecimal) representation of the following three bytes in the sequence parameter set NAL unit 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,

in bit-significance order starting from the most significant bit, and 3) level_idc. Note that reserved_zero_5bits 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.

If the profile-level-id parameter is used for indicating properties of a NAL unit stream, it indicates the profile and level that a decoder has to support in order to comply with [1] when decoding the stream. The profile-iop byte indicates whether the NAL unit stream also obeys all constraints of the indicated profiles as follows. If bit 7 (the most significant bit), bit 6, or bit 5 of profile-iop is equal to 1, all constraints of the Baseline profile, the Main profile, or the Extended profile, respectively, are obeyed in the NAL unit stream.

If the profile-level-id parameter is used for capability exchange or session setup procedure, it indicates the profile that the codec supports and the highest level that is supported for the signaled profile. The profile-iop byte indicates whether the codec has such additional limitations that only the common subset of the algorithmic features and limitations of the profiles signaled with the profile-iop byte and the profile indicated by profile_idc is supported by the codec. For example, if a codec supports only the common subset of the coding tools of the Baseline profile and the Main profile at level 2.1 and below, the profile-level-id becomes 42E015, in which 42 stands for the Baseline profile, E0 indicates that only the common subset for all profiles is supported, and 15 indicates level 2.1.

> Informative note: Capability exchange and session setup procedures should provide means to list the capabilities for each supported codec 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]).

If no profile-level-id is present, the Baseline

Profile without additional constraints at Level 1 MUST be implied.

max-mbps, 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 profile-level-id parameter MUST be present in the same receiver capability description that contains any of these parameters. The level conveyed in the value of the profile-level-id parameter MUST be such that the receiver is fully capable of supporting. max-mbps, max-fs, max-cpb, max-dpb, and max-br MAY be used to indicate such capabilities of the receiver that extend the required capabilities of the signaled level as specified below.

When more than one parameter from the set (max-mbps, 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 level with the extension of both the frame rate and bit rate is supported. That is, the receiver is able to decode such NAL unit streams in which the macroblock processing rate is up to max-mbps (inclusive), the bit rate 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 other properties comply with the level specified in the value of the profile-level-id parameter.

A receiver MUST NOT signal such values of max-mbps, max-fs, max-cpb, max-dpb, and max-br that meet the requirements of a higher level, referred to as level A herein, compared to the level specified in the value of the profile-level-id parameter, if the receiver can support all the properties of level A.

> Informative note: When the OPTIONAL MIME type parameters are used to signal the properties of a NAL unit stream, max-mbps, 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 required by the signaled level conveyed in the value of the profile-level-id parameter. When max-mbps is signaled, the receiver MUST be able to decode NAL unit streams that conform to the signaled level with the exception that the MaxMBPS value in Table A-1 of [1] for the signaled 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 for the level given in Table A-1 of [1]. Senders MAY use this knowledge to send pictures of a given size at a higher picture rate than indicated in the signaled 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 required by the signaled level conveyed in the value of the profile-level-id parameter. When max-fs is signaled, the receiver MUST be able to decode NAL unit streams that conform to the signaled level with the exception that the MaxFS value in Table A-1 of [1] for the signaled 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 for the level given in Table A-1 of [1]. Senders MAY use this knowledge to send larger pictures at a proportionally lower frame rate than indicated in the signaled 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 (see A.3.1 item i of [1]) and in units of 1200 bits for the NAL HRD parameters (see A.3.1 item j 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 level conveyed in the value of the profile-level-id parameter. When max-cpb is signaled, the receiver MUST be able to decode NAL unit streams that conform to the

signaled level with the exception that the
MaxCPB value in Table A-1 of [1] for the

signaled level is replaced with the value of max-cpb. The value of max-cpb MUST be greater than or equal to the value of MaxCPB for the level given in Table A-1 of [1]. Senders MAY use this knowledge to construct coded video streams with greater variation of bitrate compared to which 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 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 from 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 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 1024 bytes. The max-dpb parameter signals that the receiver has more memory than the minimum amount of decoded picture buffer memory required by the signaled level conveyed in the value of the profile-level-id parameter. When max-dpb is signaled, the receiver MUST be able to decode NAL unit streams that conform to the signaled level with the exception that the MaxDPB value in Table A-1 of [1] for the signaled level is replaced with the value of max-dpb. Consequently, a receiver that signals max-dpb MUST be capable of storing the following number of decoded frames, complementary field pairs, and non-paired fields in its decoded picture buffer:

Min(1024 * max-dpb / (PicWidthInMbs * FrameHeightInMbs * 256 * ChromaFormatFactor), 16)

PicWidthInMbs, FrameHeightInMbs, and ChromaFormatFactor are defined in [1].

The value of max-dpb MUST be greater than or equal to the value of MaxDPB for the level given in Table A-1 of [1]. 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, and is a property of the video decoder only. 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.

max-br: The value of max-br is an integer indicating the maximum video bit rate in units of 1000 bits per second for the VCL HRD parameters (see A.3.1 item i of [1]) and in units of 1200 bits per second for the NAL HRD parameters (see A.3.1 item j of [1]).

The max-br parameter signals that the video decoder of the receiver is capable of decoding video at a higher bit rate than required by the signaled level conveyed in the value of the profile-level-id parameter. The value of max-br MUST be greater than or equal to the value of MaxBR for the level given in Table A-1 of [1].

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

- The value of max-br replaces the MaxBR value of the signaled level (in Table A-1 of [1]).

o 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 level).

For example, if a receiver signals capability for 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 4,036,458 bits (1550000 / 384000 * 1000 * 1000).

The value of max-br MUST be grater than or equal to the value MaxBR for the signaled level given in Table A-1 of [1].

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. No assumption can be made from the value of this parameter that the network is capable of handling such bit rates at any given time. In particular, no conclusion can be drawn that the signaled bit rate 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 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, hence 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 hence a sender MAY send redundant slices. When the parameter is not present, then 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 capability 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 such 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 such sequence and picture parameter set NAL units, herein referred to as the initial parameter set NAL units, that MUST 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 the base64 [6] representation of the initial parameter set NAL units as specified in sections 7.3.2.1 and 7.3.2.2 of [1]. The parameter sets are conveyed in decoding order and no framing of the parameter set NAL units takes place. A comma is used to separate any pair of parameter sets in the list. Note that the number of bytes in a parameter set NAL unit is typically less than 10 bytes, but a picture parameter set NAL unit can contain several hundreds of bytes.

Informative Note: When several payload types are offered in the SDP Offer/Answer model, each with its own sprop-parameter-sets parameter, then the receiver cannot assume that those parameter sets do not use

conflicting storage locations (i.e.,
identical values of parameter set

identifiers). Hence, a receiver should double-buffer all sprop-parameter-sets and make them available to the decoder instance that decodes a certain payload type.

parameter-add: This parameter MAY be used to signal whether the receiver of this parameter is allowed to add parameter sets in its signaling response using the sprop-parameter-sets MIME parameter. The value of this parameter is either 0 or 1. 0 is equal to false, i.e., it is not allowed to add parameter sets. 1 is equal to true, i.e. it is allowed to add parameter sets. If the parameter is not present, its value MUST be 1.

packetization-mode: This parameter signals the properties of a RTP payload type or the capabilities of a receiver implementation. Only a single configuration point can be indicated, thus for when declaring capabilities to support more than one packetization-mode, 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 as defined in section 6.2 of RFC XXXX MUST be used. This mode is in use in standards using ITU-T Recommendation H.241 [17] (see section 12.1). When the value of packetization-mode is equal to 1, the non-interleaved mode as defined in section 6.3 of RFC XXXX MUST be used. When the value of packetization-mode is equal to 2, the interleaved mode as defined in section 6.4 of RFC XXXX 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 a NAL unit stream. It specifies the maximum number of VCL NAL units that precede any VCL NAL unit in the NAL unit stream in transmission order

and 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 deinterleaving buffer for the NAL unit 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 deinterleaving buffer that is specified in section 7.2 of RFC XXXX. It is guaranteed that receivers can perform the deinterleaving of interleaved NAL units into NAL unit decoding order, when the deinterleaving 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 4 294 967 295, inclusive.

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

deint-buf-cap: This parameter signals the capabilities of a receiver implementation, and indicates the amount of deinterleaving 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 4 294 967 295, inclusive.

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

sprop-init-buf-time: This parameter MAY be used to signal the properties of a NAL unit 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 buffer before starting decoding to recover the NAL unit decoding order from the transmission order. 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 starting 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 into the following series:
0 -1 1 0 -1 1 0 -1 1 ...

Thus, the value of sprop-init-buf-time in this example is 1 in terms of intervals of NAL unit transmission times.

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 4 294 967 295, 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 a NAL unit stream. It MUST NOT be used to signal transmitter or 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 such 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 of RFC XXXX.

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 higher cost than NALUs following the limitation.

The value of max-rcmd-nalu-size MUST be an integer in the range of 0 to 4 294 967 295, inclusive. If this parameter is not specified, no known limitation to the NALU size exists. Senders still need 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.

Encoding considerations:

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

A file format of H.264/AVC video is defined in [32]. This definition is utilized by other file formats such as the 3GPP multimedia file format (MIME type video/3gpp) [33] or the MP4 file format (MIME type video/mp4).

Security considerations:
See section 9 of RFC XXXX.

Public specification:
Please refer to RFC XXXX 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:
stewe@stewe.org

Intended usage: COMMON.

Author/Change controller:
stewe@stewe.org
IETF Audio/Video transport working group

8.2. SDP Parameters

8.2.1. Mapping of MIME Parameters to SDP

The MIME media type video/H264 string is mapped to fields in the Session Description Protocol (SDP) [5] 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 MIME subtype).
- The clock rate in the "a=rtpmap" line MUST be 90000.
- The OPTIONAL parameters "profile-level-id", "max-mbps", "max-fs", "max-cpb", "max-dpb", "max-br", "redundant-pic-cap", "sprop-parameter-sets", "parameter-add", "packetization-mode", "sprop-interleaving-depth", "deint-buf-cap", "sprop-deint-buf-req", "sprop-init-buf-time", "sprop-max-don-diff", and "max-rcmd-nalu-size", when present, MUST be included in the "a=fmtp" line of SDP.

These parameters are expressed as a MIME media type string, in the form of a semicolon separated list of parameter=value pairs.

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; sprop-parameter-
sets=Z0IACpZTBYmI,aMljiA==
```

8.2.2. Usage with the SDP Offer/Answer Model

When offering H.264 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", "packetization-mode", and, if required by "packetization-mode", "sprop-deint-buf-req". These three parameters MUST be used symmetrically, i.e. 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.

  Informative note: The requirement for symmetric use applies only for the above three parameters, and not for the other stream properties and capability parameters.

  To simplify handling and matching of these configurations, the same RTP payload type number used in the offer SHOULD also be used in the answer, as specified in [8]. An answer MUST NOT contain a payload type number used in the offer unless the configuration ("profile-level-id", "packetization-mode", and if present "sprop-deint-buf-req") is the same as in the offer.

  Informative note: An offerer, when receiving the answer, needs to compare payload types not declared in the offer based on media type (i.e. video/h264) and the above three parameters with any payload types it has already declared, in order to determine whether the configuration in question is new or equivalent to a configuration already offered.

- The parameters "sprop-parameter-sets", "sprop-deint-buf-req", "sprop-interleaving-depth", "sprop-max-don-diff", and "sprop-init-buf-time" describe the properties of the NAL unit stream that the offerer or answerer is sending for this media format configuration. This differs from the normal usage of the

offer/answer parameters: normally such parameters declare the

properties of the stream 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 the declaring entity with the same configuration, i.e. they are dependent on their source. As they apply for the configuration, rather then being bound to the payload type, the values may need to be applied to another payload type when sending.

- The capability parameters ("max-mbps", "max-fs", "max-cpb", "max-dpb", "max-br", ,"redundant-pic-cap", "max-rcmd-nalu-size") MAY be used to declare further capabilities. Their interpretation depends on the direction attribute. When the direction attribute is sendonly, then the parameters describe the limits of the RTP packets and the NAL unit stream that the sender is capable of producing. When the direction attribute is sendrecv or recvonly, then the parameters describe the limitations of what the receiver accepts.

- As specified above, an offerer needs to include the size of the deinterleaving buffer in the offer for an interleaved H.264 stream. To enable the offerer and answerer to inform each other about their capabilities for deinterleaving buffering, both parties are RECOMMENDED to include "deint-buf-cap". This information MAY be utilized when selecting the value for "sprop-deint-buf-req" in a second round of offer and answer. 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" parameter is used as described above. In addition, an answerer MUST maintain all parameter sets received in the offer in its answer. Depending on the value of the "parameter-add" parameter different rules apply: If "parameter-add" is false (0), the answer MUST NOT add any additional parameter sets. If "parameter-add" is true (1), the answerer, in its answer, MAY add additional parameter sets to the "sprop-parameter-sets" parameter. The answerer MUST also, independent of the value of "parameter-add", accept to receive a video stream using the sprop-parameter-sets it declared in the answer.

  Informative note: care must be taken when adding parameter sets not to cause overwriting of already transmitted parameter sets by using conflicting parameter set identifiers.

For streams being delivered over multicast, the following rules apply in addition.

- The stream properties parameters ("sprop-parameter-sets", "sprop-deint-buf-req", "sprop-interleaving-depth", "sprop-max-don-diff", and "sprop-init-buf-time") MUST NOT be changed by the answerer. Hence, a payload type can either be accepted unaltered, or removed.

- The receiver capability parameters "max-mbps", "max-fs", "max-cpb", "max-dpb", "max-br", and "max-rcmd-nalu-size" MUST be supported by the answerer for all streams declared as sendrecv or recvonly, otherwise one of the following actions MUST be performed: the media format is removed, or the session rejected.

- The receiver capability parameter redundant-pic-cap SHOULD be supported by the answerer for all streams declared as sendrecv or recvonly as follows: The answerer SHOULD NOT include redundant coded pictures in the transmitted stream, if the offerer indicated redundant-pic-cap equal to 0. Otherwise (when redundant_pic_cap is equal to 1), it is beyond the scope of this memo to recommend how the answerer should use redundant coded pictures.

Below are the complete lists of how the different parameters shall be interpreted in the different combinations of offer or answer and direction attribute.

- In offers and answers when "a=sendrecv", or no direction attribute is used, or in offers and answers where "a=recvonly" is used, the following interpretation of the parameters MUST be used.

  Declaring actual configuration or properties for receiving:
  - profile-level-id
  - packetization-mode

  Declaring actual properties of the stream to be sent (applicable only when "a=sendrecv" or no direction attribute is used):
  - sprop-deint-buf-req
  - sprop-interleaving-depth
  - sprop-parameter-sets
  - sprop-max-don-diff
  - sprop-init-buf-time

  Declaring receiver implementation capabilities:
  - max-mbps
  - max-fs
  - max-cpb
  - max-dpb
  - max-br
  - redundant-pic-cap
  - deint-buf-cap
  - max-rcmd-nalu-size

Declaring how Offer/Answer negotiation shall be performed:
- parameter-add

o In an Offer or Answer where the direction attribute "a=sendonly" is included for the media stream, the following interpretation of the parameters MUST be used:

Declaring actual configuration and properties of stream proposed to be sent:
- profile-level-id
- packetization-mode
- sprop-deint-buf-req
- sprop-max-don-diff
- sprop-init-buf-time
- sprop-parameter-sets
- sprop-interleaving-depth

Declaring the capabilities of the sender when it receives a stream:
- max-mbps
- max-fs
- max-cpb
- max-dpb
- max-br
- redundant-pic-cap
- deint-buf-cap
- max-rcmd-nalu-size

Declaring how Offer/Answer negotiation shall be performed:
- parameter-add

Further the following considerations are necessary:

o Parameters used for declaring receiver capabilities are in general downgradable, i.e. they express the upper limit for a sender's possible behavior. Thus a sender MAY select to set its encoder using only lower/lesser or equal values of these parameters. "sprop-parameter-sets" MUST NOT be used in a senders declaration of its capabilities, as the limits of the values that are carried inside the parameter sets are implicit with the profile and level used.

o Parameters declaring a configuration point are not downgradable, with the exception of the level part of the "profile-level-id" parameter. They express values a receiver expects to be used, and must be used verbatim on the sender side.

o When declaring sender's capabilities, and non-downgradable parameters are used in this declaration, then these parameters

express a configuration that is acceptable. 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. Hence, when multiple configuration offers are made, each offer requires its own RTP payload type associated with the offer.

- A receiver SHOULD understand all MIME parameters even if it only supports a subset of the payload formats 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 the usage of these, a second offer from the offerer is required in most cases to provide the stream properties parameters that the media sender will use. This also has the effect that the offerer needs to be able to receive this media format configuration, not only send it.

- If an offerer wishes to have non-symmetric capabilities between sending and receiving, the offerer has to offer different RTP sessions, i.e. different media lines declared as "recvonly" and "sendonly" respectively. This may have further implications on the system.

8.2.3. Usage in Declarative Session Descriptions

When offering H.264 over RTP using SDP in a declarative style as used in RTSP [30] or SAP [31], the following considerations are necessary.

- All parameters that are capable of indicating both the properties of a NAL unit stream and the capabilities of a receiver are used to indicate the properties of a NAL unit stream. For example, in this case, the parameter "profile-level-id" declares the values used by the stream, instead of capabilities of the sender. This results in that the following interpretation of the parameters MUST be used:
  Declaring actual configuration or properties:
  - profile-level-id
  - sprop-parameter-sets
  - packetization-mode
  - sprop-interleaving-depth
  - sprop-deint-buf-req
  - sprop-max-don-diff
  - sprop-init-buf-time

Not usable:
- max-mbps
- max-fs
- max-cpb
- max-dpb
- max-br
- redundant-pic-cap
- max-rcmd-nalu-size
- parameter-add
- deint-buf-cap

o A receiver of the SDP is required to support all parameters and all values of the parameters provided, or 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

A SIP Offer/Answer exchange where 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 -> Answer 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=Z0IACpZTBYmI,aMljiA==
a=rtpmap:99 H264/90000
a=fmtp:99 profile-level-id=42A01E; packetization-mode=1;
          sprop-parameter-sets=Z0IACpZTBYmI,aMljiA==
a=rtpmap:100 H264/90000
a=fmtp:100 profile-level-id=42A01E; packetization-mode=2;
           sprop-parameter-sets=Z0IACpZTBYmI,aMljiA==;
           sprop-interleaving-depth=45; sprop-deint-buf-req=64000;
           sprop-init-buf-time=102478; deint-buf-cap=128000
```

The above offer offers the same codec configuration in three different packetization formats. PT 98 represents single NALU mode, 99 non-interleaved mode, and 100 indicates the interleaved mode. In the interleaved mode case, the interleaving parameters that the offerer would use if the answer indicates support for PT 100 are also included. In all three cases the parameter "sprop-parameter-sets" conveys the initial parameter sets that are required for the answerer when receiving a stream from the offerer when this configuration (profile-level-id and packetization mode) is accepted.

Note that the value for "sprop-parameter-sets", although identical in the example above, 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=Z0IACpZTBYmI,aMljiA==,As0DEWlsIOp==,
          KyzFGleR
a=rtpmap:99 H264/90000
a=fmtp:99 profile-level-id=42A01E; packetization-mode=1;
          sprop-parameter-sets=Z0IACpZTBYmI,aMljiA==,As0DEWlsIOp==,
          KyzFGleR; max-rcmd-nalu-size=3980
a=rtpmap:100 H264/90000
a=fmtp:100 profile-level-id=42A01E; packetization-mode=2;
          sprop-parameter-sets=Z0IACpZTBYmI,aMljiA==,As0DEWlsIOp==,
          KyzFGleR; 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, while the answer indicates the same for what the answerer accepts to receive. In this case the offerer declared that it is willing to receive payload type 98. The answerer accepts this by declaring a equivalent payload type 97, i.e. it has identical values for the three parameters "profile-level-id", packetization-mode, and "sprop-deint-buf-req". This has the following implications for both the offerer and the answerer concerning the parameters that declare properties. The offerer initially declared a certain value of the "sprop-parameter-sets" in the payload definition for PT=98. However, as the answerer accepted this as PT=97, the values of "sprop-parameter-sets" in PT=98 must now be used instead when the offerer sends PT=97. Similarly, when the answerer sends PT=98 to the offerer, it has to use the properties parameters it declared in PT=97.

The answerer also accepts the reception of the two configurations that payload types 99 and 100 represents. It provides the initial parameter sets for the answerer-to-offerer direction, and buffering related parameters that it will use to send the payload types. It also provides the offerer with its memory limit for deinterleaving 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.
Please note that the parameter sets in the above example are not representing a legal operation point of an H.264 codec -- the base64 strings are only used for illustration.

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 has normally fatal results to the decoding process. Corruption could occur, for example, due to the erroneous transmission or loss of a parameter set data structure, but also due to the untimely transmission of a parameter set update. Hence, 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 stream in the payload (in-band) during an ongoing RTP session.

It is necessary 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. This section contains guidelines how principles A and B must be implemented within session control protocols, and is independent of the particular protocol used. Principle C is supported by the RTP payload format defined in this specification.

Picture and sequence parameter set NALUs SHOULD NOT be transmitted in the RTP payload unless reliable transport is provided for RTP, as a loss of a parameter set of either type likely prevents decoding of a considerable portion of the corresponding RTP stream. Thus, the transmission of parameter sets using a reliable session control protocol, i.e. usage of principle A or B above, is RECOMMENDED.

In the rest of the section it is assumed that out-of-band signaling provides reliable transport of parameter set NALUs, while in-band transport does not. 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) and 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 are present at the decoder prior to the NAL units that refer to them. Updating or adding of parameter sets can result in further problems, and therefore the following recommendations should be considered.

- When adding or updating parameter sets, principle C is vulnerable to transmission errors as described above, and therefore principle B is RECOMMENDED.

- When adding or updating parameter sets, care SHOULD be taken to ensure that any parameter set is delivered prior to its usage. 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 RECOMMEDED that a sender does not start sending NALUs requiring the updated parameter sets prior to acknowledgement of delivery from the signaling protocol.

- When updating parameter sets, the following synchronization issue should be taken into account. When overwriting a parameter set at the receiver, the sender needs ensure that the parameter set in question is not needed by any NALU present in the network or receiver buffers. Otherwise decoding using a wrong parameter set may occur. To lessen this problem, it is RECOMMENDED to either 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).

- When adding new parameter sets, previously unused parameter set identifiers are used. This avoids the problem identified in the previous paragraph. However, in a multiparty session and unless a synchronized control protocol is used, there is a risk that multiple entities try to add different parameter sets for the same identifier, which needs to be avoided.

- 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 principle 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, it is not possible to employ out-of-band parameter set transmission. In this case, parameter sets need 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 needs to be taken into

account and the loss probability should be reduced using the mechanisms discussed above.

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

To allow for maximum flexibility and best performance from the H.264 coder, it is recommended if possible to allow any sender to add its own parameter sets to be used in a session. Setting the "parameter-add" parameter to false should only be done in cases where the session topology prevents a participant to add its own parameter sets.

9. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [4], and any appropriate RTP profile (for example [18]). This implies that confidentiality of the media streams is achieved by encryption, for example through the application of SRTP [29]. Because the data compression used with this payload format is applied end-to-end, encryption may be performed after compression so there is no conflict between the two operations.

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 such pathological datagrams into the stream that are complex to decode and cause the receiver to be overloaded. H.264 is particularly vulnerable to such attacks because it is extremely simple to generate datagrams containing NAL units that affect the decoding process of many future NAL units. Therefore the usage of authentication of at least the RTP packet is RECOMMENDED, for example with SRTP [29].

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

As with any IP-based protocol, in some circumstances a receiver may be overloaded simply by the receipt of too many packets, either

desired or undesired. Network-layer authentication may be used to discard packets from undesired sources, but the processing cost of

the authentication itself may be too high. In a multicast environment, pruning of specific sources may be implemented in future versions of IGMP [19] and in multicast routing protocols to allow a receiver to select which sources are allowed to reach it.

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.

10. Congestion Control

Congestion control for RTP SHALL be used in accordance with RFC 3550 [4], and any applicable RTP profile, e.g. RFC 3551 [18]. This means that congestion control is required for any transmission over unmanaged best-effort networks.

The bit rate 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 bit rates, or the existence of non-reference pictures or sub-sequences [25] 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 if non-downgradable parameters, such as the profile part of the profile/level ID change, it becomes necessary to terminate and re-start the media stream, possibly using a different RTP payload type.

MANEs MAY follow the suggestions outlined in section 7.3 and remove certain not usable packets from the packet stream when that stream was damaged due to previous packet losses. This can help reducing the network load in certain special cases.

11. IANA Consideration

IANA is kindly requested to register one new MIME type, see section 8.1.

12. Informative Appendix: Application Examples

This payload specification is very flexible in its use, to cover the extremely wide application space that is anticipated for H.264. However, such a 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 ITU-T Recommendation H.241 Annex A

H.323-based video telephony systems that use H.264 as an optional video compression scheme are required to support H.241 Annex A [17] as a packetization scheme. The packetization mechanism defined in this Annex is technically identical with a small subset of this specification.

When operating according to H.241 Annex A, parameter sets 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, the picture parameters such as picture size or optional modes never change during the lifetime of a connection. Hence, 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. Since 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. Hence, the RTP packet stream consists basically of NAL units that carry single coded slices.

The encoder chooses the size of coded slice NAL units such 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 designed using RFC 2429 packetization [12]. It is also implemented and good results were reported [14].

The VCL encoder codes the source picture such 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 into another STAP. 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, because the high number of individual slices (18 for a 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 [20]. Coding individual slices according to this packetization scheme provides a 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 to several RTP packets with only one NAL unit that are preferred in a wireless network and vice versa.

12.4. Video Telephony, with Data Partitioning

This scheme is implemented and was shown to offer good performance especially at higher packet loss rates [14].

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, i.e., 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 2733 [21], for example, specifies which media packets are associated with the FEC packet.

In all cases, the incurred overhead is substantial, but in the same order of magnitude as the number of bits that have otherwise be spent for intra information. However, this mechanism is not adding 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 is implemented and was shown to provide good performance especially at higher packet loss rates [22].

The most efficient means to combat packet-losses for scenarios where retransmissions are not applicable is forward error correction (FEC). Although the application layer, end-to-end use of FEC is often less efficient when compared to 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 2733 [21] 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, i.e., n-k parity packets are generated for k information packets.

When using a code with parameters (n,k) within the RFC 2733 framework, the following properties are well-known:

a) If applied over one RTP packet, RFC 2733 provides only packet repetition.
b) RFC 2733 is most bit-rate 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 is, the smaller the residual error probability becomes. For example, for packet loss probability 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 is, 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 2733 in combination with H.264 baseline coded video without using FUs several options might be considered:

1) The video encoder produces NAL units where each video frame is coded in a single slice. Applying FEC, one could use a simple code, e.g. (n=2, k=1), i.e., each NAL unit would basically just be repeated. The disadvantage is obviously the bad code performance according to (d) and the low flexibility as only (n, k=1) codes can be used.
2) The video encoder produces NAL units where 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 the more RTP packets per frame are used. Packets of completely different length might also be connected, which decreases bit-rate efficiency according to (b). However with some care and for slices of 1kb 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, where a certain frame contains k slices of possibly almost equal length. Then, applying FEC, a better code, e.g. (n=24, k=12), over the sequence of NAL units for each frame can be used. The delay compared to (2) may be reduced, but several disadvantages are obvious. Firstly, the coding efficiency of the encoded video is lowered significantly as slice-structured coding reduces intra-frame prediction and additional slice overhead is necessary. Secondly, 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, and therefore FEC with a reasonable k and n can be applied even if the encoder made no effort of producing 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 the case 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 for achieving 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 addition, 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 bit-rate efficient use of RFC 2733.

In case that the error probability depends on the length of the transmitted packet, e.g. in case of mobile transmission [16], the benefits of applying FUs with FEC are even more obvious. Basically,

the flexibility of the size of FUs allows applying appropriate FEC for each NAL unit and even unequal error protection of NAL units.

The incurred overhead when using FUs and FEC is substantial, but 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 [22] it was shown that the overall performance at the same error rate and the same overall bit-rate including the overhead, the FEC-based approach can enhance the quality.

12.6. Low-Bit-Rate Streaming

This scheme has been implemented with H.263 and non-standard RTP packetization and gave good results [23]. 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 bit-rates 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-bit-rate video, the use of large packets means that sometimes up to few pictures should be encapsulated in one packet.

However, loss of a packet including many coded pictures would have drastic consequences in visual quality, as there is practically no other way to conceal a loss of an entire picture than to repeat the previous one. One way to construct relatively large packets and maintain possibilities for successful loss concealment is to construct MTAPs that contain slices from several pictures in an interleaved manner. 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 succeed relatively well.

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 [24]. 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, but rather it typically buffers the incoming data for a few seconds. This buffering helps to maintain continuous playback, because, 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 perfectly recovered. 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, because 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 [25]. 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 to 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 is from left to right 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 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:

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

The video communication system used in the 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 rate control purposes, for example. In some cases the pre-encoding buffer may not exist, but rather 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:

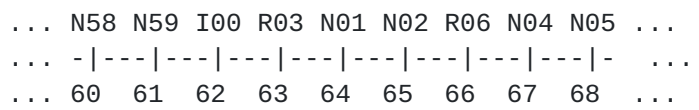


Figure 17. Re-ordered pictures in the pre-encoding buffer

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 equal to 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 MIME type parameter is set to 0, because the transmission (or reception) order is identical to the decoding order.

The decoder has to buffer for one picture interval initially 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 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, i.e., 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 MIME 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. Since 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 the following:

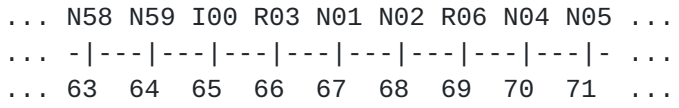


Figure 21. Interleaving: Receiver Buffer

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

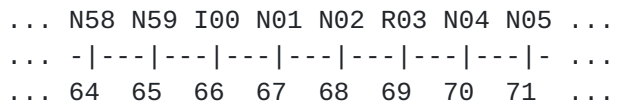


Figure 22. Interleaving: Receiver buffer after reordering

It can be observed 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 were 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 utilizing the redundant codec picture feature include the video redundancy coding [26] and protection of "key pictures" in multicast streaming [27].

One property of many error-prone video communications systems is that transmission errors are often bursty and therefore they may affect more than one consecutive transmission packets in transmission order. In low bitrate video communication it is relatively common that an entire coded picture can 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. In order to make the transmission scheme more tolerant of bursty transmission errors, it is beneficial to transmit a primary coded picture further apart from the corresponding redundant coded pictures. 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, because 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 [28] 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 consist of all NAL units that are 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. Acknowledgements

The authors thank Roni Even, Dave Lindbergh, Philippe Gentric, Gonzalo Camarillo, Joerg Ott, and Colin Perkins for careful review.

15. Full Copyright Statement

16. Intellectual Property Notice

17. References

17.1. Normative References

[1] ITU-T Recommendation H.264, "Advanced video coding for generic audiovisual services", May 2003.
[2] ISO/IEC International Standard 14496-10:2003.
[3] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[4] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
[5] M. Handley and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.
[6] S. Josefsson, "The Base16, Base32, and Base64 Data Encodings", RFC 3548, July 2003.
[7] ITU-T Recommendation T.35, "Procedure for the allocation of ITU-T defined codes for non-standard facilities", February 2000.
[8] J. Rosenberg, and H. Schulzrinne, "An Offer/Answer Model with the Session Description Protocol (SDP)", RFC 3264, June 2002.

17.2. Informative References

[9] "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)", available from ftp://ftp.imtc-files.org/jvt-experts/2003_03_Pattaya/JVT-G050r1.zip, May 2003.
[10] A. Luthra, G.J. Sullivan, and T. Wiegand (eds.), Special Issue on H.264/AVC. IEEE Transactions on Circuits and Systems on Video Technology, July 2003.
[11] P. Borgwardt, "Handling Interlaced Video in H.26L", VCEG-N57r2, available from http://ftp3.itu.int/av-arch/video-site/0109_San/VCEG-N57r2.doc, September 2001.
[12] C. Borman et. Al., "RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H.263+)", RFC 2429, October 1998.
[13] ISO/IEC IS 14496-2.
[14] S. Wenger, "H.26L over IP", IEEE Transaction on Circuits and Systems for Video technology, July 2003.
[15] S. Wenger, "H.26L over IP: The IP Network Adaptation Layer", Proceedings Packet Video Workshop 02, April 2002
[16] T. Stockhammer, M.M. Hannuksela, and S. Wenger, "H.26L/JVT Coding Network Abstraction Layer and IP-based Transport" in Proc. ICIP 2002, Rochester, NY, September 2002.
[17] ITU-T Recommendation H.241, "Extended video procedures and control signals for H.300 series terminals", 2004.
[18] H. Schulzrinne and S. Casner, "RTP Profile for Audio and Video

Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

[19] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

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

[21] J. Rosenberg, H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction", RFC 2733, December 1999.

[22] T. Stockhammer, T. Wiegand, T. Oelbaum, 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.

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

[24] S.H. Kang and A. Zakhor, "Packet scheduling algorithm for wireless video streaming," International Packet Video Workshop 2002, available http://www.pv2002.org.

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

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

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

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

[29] Baugher, McGrew, Carrara, Naslund, and Norrman, "The Secure Real-time Transport Protocol," RFC 3711, Internet Engineering Task Force, March 2004.

[30] H. Schulzrinne, A. Rao, R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, Internet Engineering Task Force, April 1998.

[31] M. Handley, C. Perkins, E. Whelan, "Session Announcement Protocol", RFC 2974, Internet Engineering Task Force, June 2001.

[32] ISO/IEC 14496-15: "Information technology - Coding of audio-visual objects - Part 15: Advanced Video Coding (AVC) file format".

[33] D. Singer, and R. Castagno, "MIME Type Registrations for 3GPP Multimedia files", Internet Draft, draft-singer-avt-3gpp-mime-01, Sep 2003.

Author's Addresses

Stephan Wenger Phone: +49-172-300-0813

TU Berlin / Teles AG Email: stewe@stewe.org
Franklinstr. 28-29

D-10587 Berlin
Germany

Miska M. Hannuksela
Nokia Corporation
P.O. Box 100
33721 Tampere
Finland

Phone: +358-7180-73151
Email: miska.hannuksela@nokia.com

Thomas Stockhammer
Institute for Communications Eng.
Munich University of Technology
D-80290 Munich
Germany

Phone: +49-89-28923474
Email: stockhammer@ei.tum.de

Magnus Westerlund
Multimedia Technologies
Ericsson Research EAB/TVA/A
Ericsson AB
Torshamsgatan 23
SE-164 80 Stockholm
Sweden

Phone: +46-8-7570000
Email:
magnus.westerlund@ericsson.com

David Singer
QuickTime Engineering
Apple
1 Infinite Loop MS 302-3MT
Cupertino
CA 95014
USA

Phone +1 408 974-3162
Email: singer@apple.com

18. RFC Editor Considerations

The RFC editor is requested to remove this section before publications as a RFC. The RFC editor is also requested to replace all occurrences of XXXX with the RFC number this document receive.

If available at the time of publication please do update reference 33 with the assigned RFC number.