

H265 (HEVC) over 1722 proposal

Presented to 1722 Working Group on 9th June 2020

Kieran Tyrrell - Sienda

Suggestion/direction to editor: [blue text](#)

Text where proposed H265 section differs from existing H254 section: [yellow highlight](#)

[Add row to Table 20, and adjust last row to decrease reserved range:](#)

Table 20 - format_subtype field for RFC Format

Value	Name	Description	Clause
0 ₁₆	MJPEG	MJPEG Format (RFC 2435)	8.4
1 ₁₆	H264	H.264 Format (RFC 6184)	8.5
2 ₁₆	JPEG2000	JPEG 2000 Video (RFC 5371)	8.6
3 ₁₆	H265	H.265 Format (RFC 7798)	8.7
4 to FF ₁₆	Reserved	Reserved	-

[Add section 8.7 as follows, and renumber Figures. Section 8.7 is very similar to section 8.5. Differences are \[highlighted\]\(#\).](#)

8.7 H.265 Compressed Video Format

8.7.1 Overview

The [H.265](#) Compressed Video Format uses a format value of RFC (see Table 19) and a format_subtype value of [H265](#) (see Table 20).

This format uses the video_data_payload field to define the following fields:

- [h265_timestamp](#): 4 octets
- [h265_payload](#): 0 to n octets

The fields of the [H.265](#) Compressed Video Format are given in Figure [41](#).

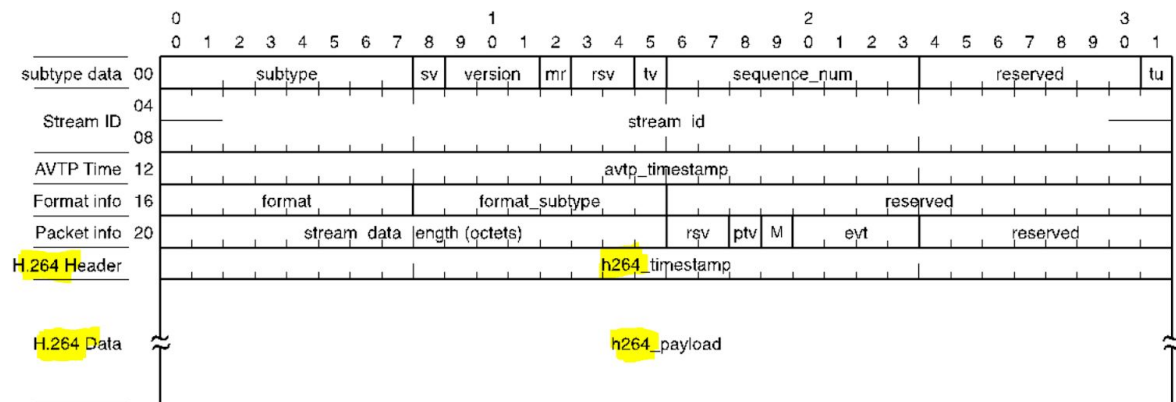


Figure 41 - H.265 Compressed Video Format PDU
(edit h264 to h265, the structure is the same)

8.7.2 Common header data encapsulation

8.7.2.1 Overview

The **H.265** video format uses the defined Compressed Video Format header fields as defined in Figure 38, with specific usage defined in this subclause.

8.7.2.2 avtp_timestamp field

The avtp_timestamp represents the AVTP Presentation Time associated with the given **H.265** Network Abstraction Layer (NAL) unit. It is based on the gPTP time at which the **H.265** Talker receives the **H.265** NAL unit and is calculated according to the techniques described in 4.4.4.9. The Talker shall fill the avtp_timestamp with this AVTP Presentation Time, but only in the **H.265** AVTPDU carrying the final fragment of the video packetization unit. **H.265** AVTPDUs carrying pre-final video packetization unit fragments shall be marked with an invalid avtp_timestamp by setting the timestamp_valid field to zero (tv = 0).

NOTE—The avtp_timestamp field does not carry the timing information produced during the **H.265** encoding process and that is associated with decoding and rendering/presentation of each **H.265** NAL unit. That timestamp is carried in the **h265_timestamp** field (see 8.7.3.1).

8.7.2.3 pty field

The pty field indicates the validity of the payload timestamp field. In the case of the **H.265** AVTPDU, it indicates the validity of the **h265_timestamp** field.

The H.265 CVP Talker shall set the ptv bit to one (1) if the h265_timestamp field contains a valid value, or zero (0) otherwise.

The H.265 CVP Listener shall ignore the value of the h265_timestamp field if ptv is set to zero (0).

NOTE—H.265 CVP Talkers are encouraged to insert valid H.265 timestamps to provide valuable assistance to the H.265 decoder, eliminate the need for the decoder to recreate NAL unit timestamps, and thereby reduce decoding complexity and latency. Often NAL unit timestamp information is available from the H.265 encoding process; H.265 CVP Talkers should make every effort to transmit this information in the H.265 AVTPDU.

8.7.2.4 M field

The M field shall be set according to RFC7798, Section 4.1.

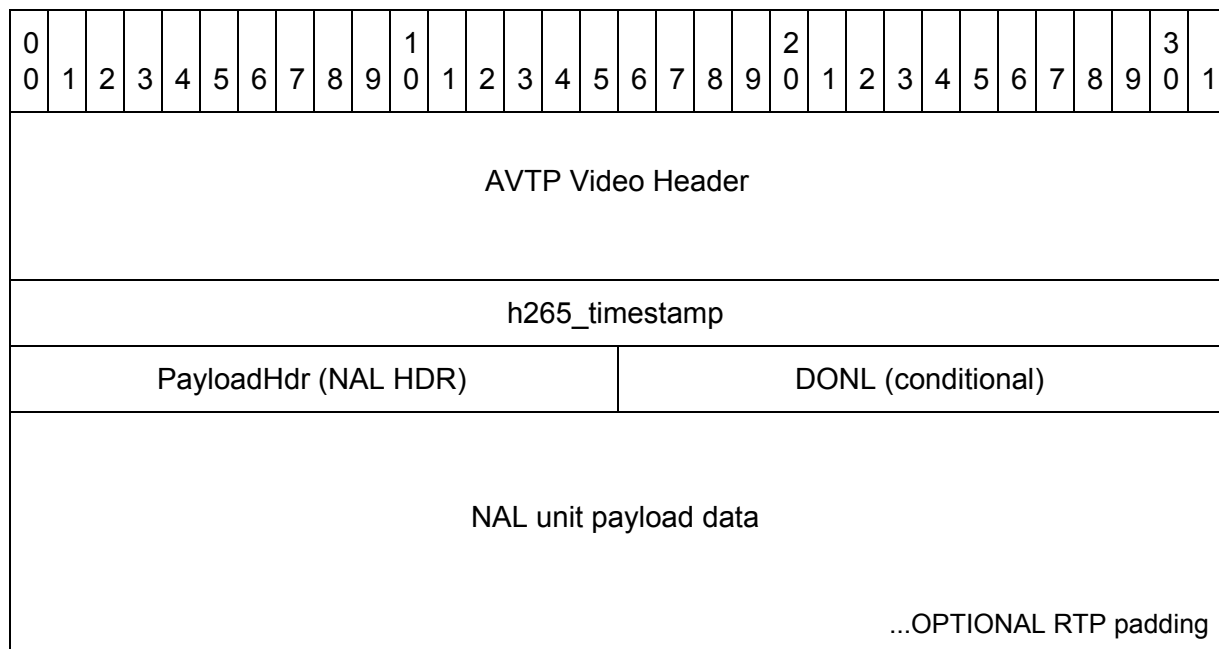
8.7.3 Additional header and data fields

8.7.3.1 h265_timestamp field

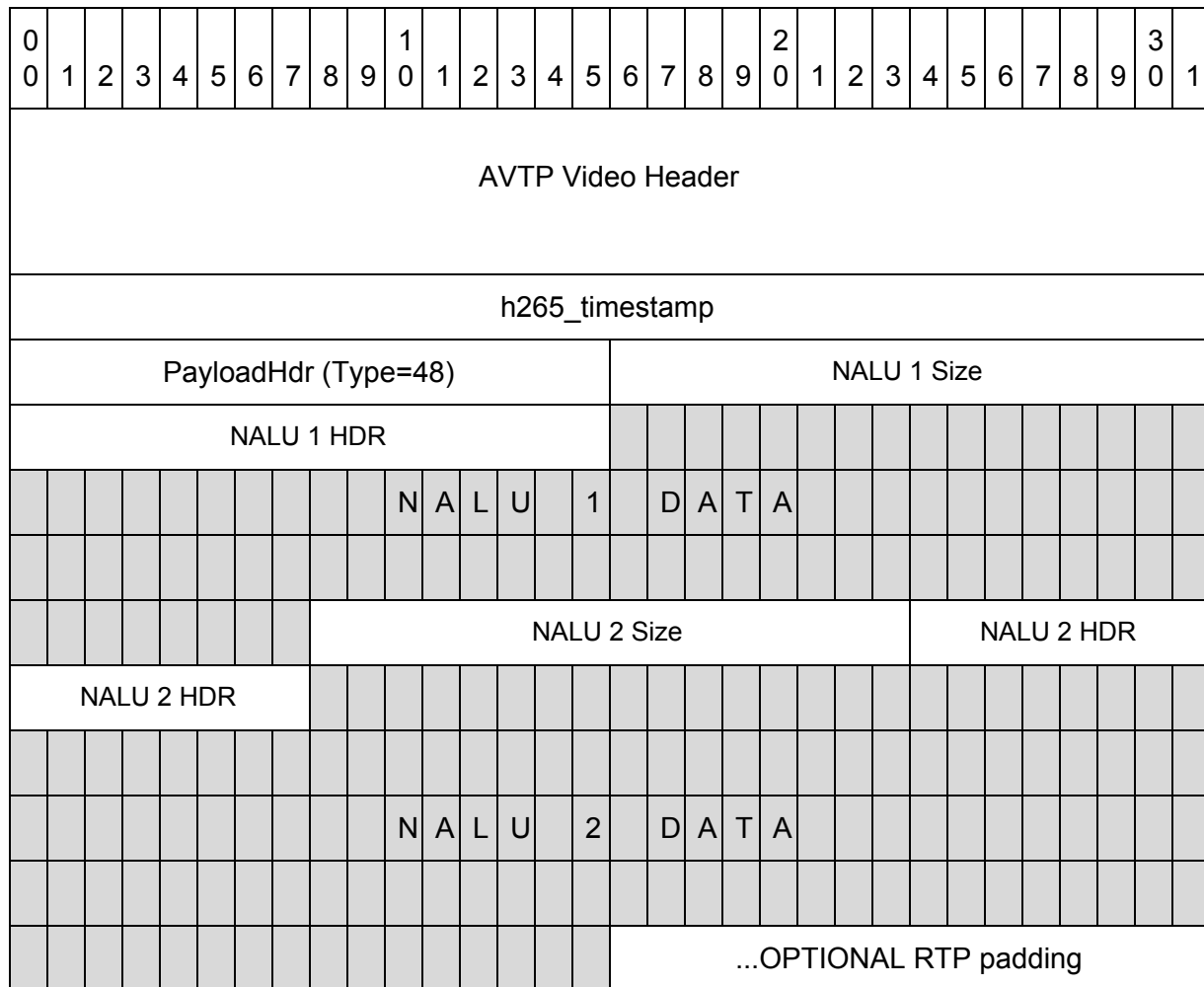
The h265_timestamp field carries the timestamp information associated with the H.265 NAL unit fragment. It is used by the H.265 decode process to properly synchronize, decode, and present H.265 video frames. The h265_timestamp shall be calculated according to RFC7798, Section 4.1.

8.7.4 H.265 payload

The h265_payload field shall be constructed according to RFC7798 Sections 4.4 through 5.8, 5, and 6. This includes payloads that are constructed of a single H.265 NAL unit (RFC7798, Section 4.4.1), of an aggregation of H.265 NAL units (RFC7798, Section 4.4.2), and of a fragmentation of a large H.265 NAL unit (RFC7798, Section 4.4.3). Examples of these AVTPDUs are shown in Figure 42, Figure 43, Figure 44, Figure 45, Figure 46, and Figure 47. In each of these diagrams, the AVTP Video Header block represents all of the header fields shown in Figure 41, from the subtype field to the reserved field at octet offset XX?, inclusively. The h265_timestamp field is explicitly shown in each figure.



Example of an AVTP packet including a single NAL Unit payload (NOTE: a h264 equivalent of this is NOT currently represented in 1722. Do we need this? It is almost a clone of RFC7798 Section 4.4.1 Figure 3.



Example of an AVTP packet containing two aggregation units without the DONL and DOND fields (note this is similar to RFC7798 Section 4.4.2 Figure 7 but with AVTP header instead of RTP header)

Create Figures for examples of AVTP packets similar to examples in RFC7798 Section 4.4.2-3 containing:

- Two Aggregation Units with the DONL and DOND Fields (Figure 8 from RFC7798)
- Fragmentation
- PAYload Content Information (PACI)? (Section 4.4.4)
- Temporal Scalability Control Information? (Section 4.5)

Next steps...

- Should we increase the size of the sequence number field??!
- Check [4.3](#). Transmission Modes:
 - o a Single RTP stream on a Single media Transport (SRST),
 - o Multiple RTP streams over a Single media Transport (MRST), or
 - o Multiple RTP streams on Multiple media Transports (MRMT).