



TSD

Discussion about the AVTP timestamp for H.264 video transmission described in 1722_D12 chapter 9.5.

Christian Sørensen



Agenda

- Development Setup
- Usage and calculation of the AVTP H.264 timestamp not clear
- RFC 6184 basics (packetization types, usage of PTS)
 - AVB / RTP differences
 - AVTP timestamp vs RTP timestamp
 - Which timing information is needed for a H264 decoder ?

Development Setup

- The H.264 Video format described in chapter 9 in IEEE 1722-D12 is mainly based on RFC6184 (RTP payload format). Only the header and the presentation time handling is different. Idea was to use the gstreamer to create RTP packets and doing the AVTP header re-ordering behind.
- Gstreamer pipelines:
AVB Transmitter: filesrc ! qtdemux ! rtpH264pay ! avbsink -> AVB Stack
AVB Receiver: AVB Stack -> avbsrc ! rtpjitterbuffer ! rtpH264depay ! vaapidcode ! vaapisink
- Problems:
 - Which time should be used for avtp_timestamp calculation described in 1722-D12 5.4.3.8
 - PTS or RTP timestamp ?
 - How to create the AVTP timestamp from the RTP timestamp and vice versa ?
 - Is it possible at all ?
 - Could the RTP timestamp used for transmission ?
 - Do we have to know the PTS and the DTS for calculation ? That is not given by using the output from rtpH264pay.
 - The RTP timestamp is not increasing monotonically by constantly increasing sequence numbers (see explanation on the next foils)

RFC 6184 basics

- RFC6184: 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 .
 - PTS and DTS are *not* part of the H.264 bitstream.
 - There is no timing information carried within the NAL units which can be used for the decoder. It is mandatory to have the RTP timestamp for this. So the timestamp transmitted via the AVB header couldn't be used as usual for AVB. The decoder will restore the DTS and the PTS by getting the RTP timestamp and knowing the ordering scheme.
- RTP timestamp and PTS timestamp differ in encoding and definition!
- RFC 6184: Receivers SHOULD use the RTP timestamp for synchronizing the display process.

RFC 6184 basics

- Packetization Modes used by RTP H.264 (RFC6184: **5.4. Packetization Modes**)
 - Single NAL unit mode
 - Non-interleaved mode (In the non-interleaved mode, NAL units are **transmitted in NAL unit decoding order**. It is used for low latency streaming)
 - Interleaved mode: (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.)
- In case of the „ Non-interleaved mode “ the RTP timestamps are not increasing constantly.

“The semantics of the RTP timestamps depends on the payload type. As a general rule it contains the Presentation Time Stamp (PTS) of the first block of data inside the packet. Also as a general rule, the packets are ordered by Decoding Time Stamp (DTS). In the presence of bi-directional predicted frames ("B-frames"), the codec would logically send the later key frame ("I-frame") first, as it should have an earlier DTS inspite of the later PTS. It is hence perfectly normal to have a late timestamp before an earlier one, within short time interval.I do not think there is a bug anywhere.” (Source: <https://mailman.videolan.org/pipermail/vlc-devel/2009-March/057179.html>)
- So interleaved as non-interleaved packetization mode could be possible we have to deal with increasing presentation timestamps for interleaved mode and not monotonically increasing presentation timestamps for non-interleaved mode.

Summary

- In case of not constantly increasing timestamps the usual AVTP timestamp usage is changed.
 - The model of AVTP Timing Reference Planes as shown in 1722-D12 Figure 5.1 is not valid anymore!
 - The presentation timestamp does not anymore define the timing to hand over the packet to the application.
 - The timestamp as is defined in RFC6184 completely differs the usage from an AVTP_timestamp.
 - This timestamp can't be used to derive the transmission timing!
- Proposed solution:
 - Use the class observation interval to send out the packets
 - Use the timestamp on receiver side in same manner as described in RFC6184
 - Use the rtp timestamp as defined in RFC6184 or specify a dependency between the h264 rtp timestamp and the h264 AVB timestamp.