3GPP TSG SA WG4 eMeeting #123S4-230529

17th – 21st April 2023

**Source:** Samsung Electronics Co., Ltd.

**Title: [5G\_RTP] RTP header extension for pose information**

**Document for** Discussion and Agreement

# 1. Introduction

In the last 122nd meeting, S4-230359 [1] and S4-230390 [2] are discussed and agreed in 5G\_RTP and MeCAR WI discussions respectively. The RTP header extension for rendered pose and timestamp are described in [1] while various timestamps capturing events occurred in device and split rendering function are described in [2].

This contribution proposes the merged text to adopt the agreed content in MeCAR for 5G\_RTP. More specifically, the timestamps estimated or measured from device (e.g., estimated-at-time, estimated-target-display-time, sent-at-time, previous-render-to-photon-time), and from the split rendering function (e.g., start-to-render-at-time, estimated-target-display-time) are proposed.

Semantic of the timestamps can be referred to clause 8.3 of MeCAR PD v5.0 as enclosed:

[Start of quote]

## 8.3 Pose information

The MeCAR device sends a group of pose information to the server's split render function to generate rendered media frames based on the poses. Each pose is associated with time metadata, such as the time when the pose estimation was made (T1), the estimated target display time of the content (T2.estimated), and the time the group of poses was sent (T1').

The gap between the actual-target-display-time (T2.actual) and the pose estimate time (T1) is the pose-to-render-to-photon delay, which allows the MeCAR device to know the amount of processing time as well as the connection delay required for a loop of split rendering. The next round of pose estimation should refer to the pose-to-render-to-photon delay for the estimation of a new T2.estimated.

The split render function in the server may refer to T1', which is the time when the group of poses is sent from device, if multiple pairs of pose and metadata for the same target display time are received from the device. The T1’ information may be used to manage poses by the server, such as allowing the MeCAR device to update former estimations by resubmitting a new pose with the same estimated-target-display-time.

The split render function in the server sends rendered media frames and associated metadata. The metadata shall include the pose used for the rendered frame, as well as corresponding time information, such as T1, T2.estimated, and may include the time when the rendering started (T3), all sent to the MeCAR device in order to measure the render-to-photon delay.

In the case where there are poses stacked in the server’s pose buffer, for example with a granularity finer than the device's supported frame rate, the split render function should select the pose closest to the display time, according to the previous-render-to-photon delay. The previous-render-to-photon delay from the most recent frame information may help the server to make this selection.



* Pose and timestamp information from the device
  + Estimated pose
  + Estimated-at-time (T1)
  + Estimated-target-display-time (T2.estimated)
  + Sent-at-time (T1')
  + Previous-render-to-photon-time (T2.actual-T3)
* Pose and timestamp information associated with rendered media frame from the server
  + Pose used for rendering
  + Estimated-at-time (T1)
  + Estimated-target-display-time (T2.estimated)
  + Start-to-render-at-time (T3)

Pose includes location and direction information.

Estimated-at-time (T1) is the actual time of when the pose estimation was made.

Estimated-target-display-time (T2.estimated) is the estimated target display time for the media frame which is rendered, or will be rendered, using to this pose.

Sent-at-time (T1') is the actual time when a pose or a group of poses is sent from the device to the server.

Previous-render-to-photon-time (T2.actual.previous-T3.previous) is the render-to-photon delay for the most recent frame.

Start-to-render-at-time (T3) is the actual time when the renderer in the split rendering server starts to render the associated media frame.

[End of quote]

# 2. Proposed changes

[Change 1]

## 4.3 RTP Header Extension for Rendered Pose

The split rendering server streams the rendered frame using one or more video streams, depending on the view and projection configuration that is selected by the UE. The server uses the proposed RTP header extension to associate the selected pose with the rendered frame. An RTP header extension is the most appropriate option to associate the rendered frame with its pose as it is carried as part of the RTP packets that carry the rendered images of a frame. The RTP header extension may also be used with audio streams of a split rendering process. The frequency of RTP header extension for pose is once in a frame/PDU set. It may be sent multiple times but not necessarily in every RTP packet.

Header extensions are declared in the SDP using the “a=extmap” attribute as defined in RFC8285. The header extension is identified through an association between the URI of the header extension and an ID value that is contained as part of the extension. The rendered pose header extension should use the following URN: “urn:3gpp:xr-rendered-pose”.

The two-byte header format of the header extension is used for signaling the rendered pose. The 2-byte (RFC8285) RTP header extension format of the rendered pose header extension is as follows:

0 1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+| 0x100 | appbits | ID | length=44+2n. |+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+| x |+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+| y |+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+| z |+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+| rx |+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+| ry |+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+| rz |+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+| rw |+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+| || estimated-at-time (T1) |+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+| || start-to-render-at-time (T3) |+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+| action\_id #1 | ... |+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

The (x,y,z) provides the position of the rendered pose and the (rx,ry,rz,rw) provides the orientation of the rendered pose.

Estimated-at-time(T1) and start-to-render-at-time(T3) provides the time when pose was estimated and when the split rendering server started to render the rendered frame.

These timestamps use the XR system clock. There is no requirement to synchronize the RTP stream timestamps to the XR system clock. The timestamps are passed to the XR runtime together with the rendered swapchain images (e.g. as part of the xrEndFrame call in OpenXR).

Alternatively to this format, the application and the rendering server may use unique identifiers for the transmitted pose information to reduce the required extension header size.

The header also provides the identifiers of all actions that were processed for the rendering of the frame. A maximum limit of 10 actions per RTP header extension for rendered pose is allowed. Hence, the total size of the header extension is set to 4 (7 + 2 2) + 2 n, where m is the number of timestamps and n is the number of action identifiers in the header extension.

[End of change 1]

# 3. Proposal

It is proposed to adopt the proposed change in clause 2 to section 4.3 of 5G\_RTP PD v.0.0.4.

# 4. References

[1] S4-230359, "Signaling the render pose and other related information"

[2] S4-230390, "[MeCAR] On pose information"