**3GPP TSG-SA WG4 Meeting #127S4-240222Rev1**

**Sophia-Antipolis, France, 29 January - 2 February 2024**

**Source: Interdigital Finland Oy**

**Title: [SR\_MSE] QoE metrics timing information format**

**Spec: 3GPP TS 26.565 v1.0.0**

**Agenda item: 8.6**

**Document for: Agreement**

**1. Introduction**

At meeting SA4#126, the definition of latency metrics was agreed. The latency metrics are measured using timing information that were collected at different stages of the rendering process in the Split Rendering Client and the SR Server.

The server sends to the client the collected timing information with the RTCP report block format. The Timing Information Format in clause 9.3.2.1 lists the timing information associated with the rendered frame that is transmitted in the RTCP report block formats.

However, an editor’s note remains in clause 9.3.2.1 about the reference time used for the timing information is for FFS.

**2. Reason for Change**

In this contribution We propose to clarify the time format of the timing information to address the editor’s note in clause 9.3.2.1: “the reference time used for the timing information is for FFS.”

* 1. **Time formats of the timing information**

The timing information is collected on different blocks in the Split rendering Client and Server to measure the latency metrics.

However, the timing information is in different time format.

**Timing information coming from the XR runtime:**

* lastChangeTime: Time of the user action.
* actualDisplayTime (ref. T2.actual): The actual display time of the rendered frame in the swapchain.
* estimatedAtTime (ref. T1): The time at which the pose was estimated.

This timing information is expressed in XR system time clock.

With the Kronos OpenXR API, the XR system time is a XrTime that can be converted to monotonic clock time using the xrConvertTimeToTimespecTimeKHR. The xrConvertTimeToTimespecTimeKHR function converts an XrTime to time as if generated by clock\_gettime with CLOCK\_MONOTONIC.

estimatedAtTime, lastChangeTime are sent from the client to the server. The server sends it back to the client as is, along with the associated media frame using the RTCP report block format.

**Timing information from the client:**

* sendingAtTime (ref. T1’): The time when the metadata message is transmitted. It is sent from the client to the server. It is then received by the server and sent back without change to the client with the associated media frame.
* receptionTime: The time when the data is received by the split rendering client.

The timing information can be expressed as NTP timestamp. During transmission time to the client in the RTCP report block format, it can be translated into RTP timestamps. Once received by the client, it can be translated back to NTP timestamp format.

**Timing information from the server:**

* startToRenderAtTime (ref. T3): The time at which the renderer starts to render the scene.
* sceneUpdateTime (ref. T6): The time at which the Scene Manager starts to update the 3D scene graph according to the viewer pose and the user actions.
* serverTransmitTime (ref. T5): The time at which the encoded rendered frame is transmitted from the SRS to the SRC.

The timing information collected on the server side can be expressed as NTP timestamp.

During transmission time to the client in the RTCP report block format they can be translated into RTP timestamps. Once received by the client, they can be translated back to NTP timestamp format.

**2.2 Latency computation**

The latency metrics are specified in Table 9.3.2-2. Latency calculation formulas are defined using the above timing information.

Some of the latency metrics, listed below, are computed using timing information expressed in different time formats.

|  |  |
| --- | --- |
| renderToPhoton | The time duration, in units of milliseconds, between the start of the rendering by the Presentation Engine and the display time of the rendered frame. It can be computed as follows: actualDisplayTime – startToRenderAtTime  Mixing XR system time clock and NTP timestamp. |
| userInteractionDelay | The time duration, in units of milliseconds, between the time a user action is initiated and the time the action is taken into account by the content creation engine in the scene manager. It can be computed as follows: sceneUpdateTime – lastChangeTime  Mixing NTP timestamp and XR system time clock. |
| ageOfContent | The time duration, in units of milliseconds, between the time the content is created in the scene by the Scene Manager and the time it is presented to the user. It can be computed as follows: actualDisplayTime – sceneUpdateTime  Mixing XR system time clock and NTP timestamp. |

For the above latency computation, the timing information mentioned need to be converted to a single time formats. For example, the system wall clock time can be used for this purpose.

It is left to the implementation to perform the time conversion to calculate above latencies.

**3. Proposal**

It is proposed to agree the following changes to 3GPP TS 26.565.

\* \* \* First Change \* \* \* \*

### 9.3.2 QoE Metrics Formats

#### 9.3.2.1 Timing Information Format

The timing information associated with the rendered frame is transmitted in the RTCP report block format. This timing information is listed in the Table 9.3.2.1-1.

The latency metrics that use the timing information defined in Table 9.3.2.1-1 are detailed in the section 9.3.2.2.

Table 9.3.2.1-1: Timing information in the RTCP block formats.

|  |  |
| --- | --- |
| **Name** | **Description** |
| estimatedAtTime | This wall clock time is defined in Table 8.2.2.2-1 - Pose  Format.This time is sent from the split rendering client.  This time is then received by the split rendering server and sent back to the split rendering client with the associated media frame. |
| sendingAtTime (ref. T1’) | This time is defined in Table 8.2.3-2 - Split Rendering Metadata Message Data Type  This time is sent from the split rendering client.  This time is then received by the split rendering server and sent back to the split rendering client with the associated media frame. |
| startToRenderAtTime (ref. T3) | The time when the renderer in the Split Rendering Server starts to render the associated media frame. |
| sceneUpdateTime (ref. T6) | The time when the Scene Manager starts to update the 3D scene graph according to the viewer pose and the user actions. |
| serverTransmitTime (ref. T5) | The time when the encoded rendered frame is transmitted from the split rendering server to the split rendering client. |

9.3.2.2 Latency metrics

To enable good XR experiences, it is relevant to monitor latencies such as the motion-to-photon and the pose-to-render-to-photon.

Beyond the sense of presence and immersiveness, the age of the content and user interaction delay are of the uttermost importance for immersive and non-immersive interactive experiences, i.e., experiences for which the user interaction with the scene impacts the content of scene (such as online gaming).

Table 9.3.2-1 provides timing information collected to compute the latency metrics at the split rendering client or split rendering server endpoint.

*Table 9.3.2-1: Timing information for latency metrics*

|  |  |  |
| --- | --- | --- |
| **Source endpoint** | **Timing information** | **Definition** |
| Split Rendering Client | estimatedAtTime (ref. T1) | Ref. Table 9.3.2.1-1  This time is expressed in XR system time clock. |
| Split Rendering Client | lastChangeTime | The time the user action is made. It corresponds to the lastChangeTime field defined in the action format in Table 9. This time is expressed in XR system time clock. |
| Split Rendering Server | sceneUpdateTime (ref. T6) | Ref. Table 9.3.2.1-1  This time is a NTP timestamp format. |
| Split Rendering Server | startToRenderAtTime (ref. T3) | Ref. Table 9.3.2.1-1.  This time is a NTP timestamp format. |
| Split Rendering Client | actualDisplayTime (ref. T2.actual) | The actual display time of the rendered frame in the swapchain. The estimation of the actual display time is available through the XR runtime. This time is expressed in XR system time clock. |
| Split Rendering Client | sendingAtTime (ref. T1’) | Ref. Table 9.3.2.1-1.  This time is a NTP timestamp format. |
| Split Rendering Server | serverTransmitTime (ref. T5) | Ref. Table 9.3.2.1-1.  This time is a NTP timestamp format. |
| Split Rendering Client | receptionTime | The time when the data is received by the split rendering client. This time is a NTP timestamp format. |

The latency metrics are specified in Table 9.3.2-2. Latency calculation formulas are defined using the timing information defined in Table 9.3.2-1.

*Table 9.3.2-2: Latency metrics*

|  |  |
| --- | --- |
| **Latency metric** | **Description** |
| poseToRenderToPhoton | The time duration, in units of milliseconds, between the time to provide the pose information from the XR runtime to the renderer (the renderer uses this pose to generate the rendered frame) and the display time of the rendered frame. It can be computed as follows: actualDisplayTime – estimatedAtTime |
| renderToPhoton | The time duration, in units of milliseconds, between the start of the rendering by the Presentation Engine and the display time of the rendered frame. It can be computed as follows: actualDisplayTime – startToRenderAtTime (NOTE 1) |
| roundtripInteractionDelay | The time duration, in units of milliseconds, between the time a user action is initiated and the time the action is presented to the user. It can be computed as follows: actualDisplayTime – lastChangeTime |
| userInteractionDelay | The time duration, in units of milliseconds, between the time a user action is initiated and the time the action is taken into account by the content creation engine in the scene manager. It can be computed as follows: sceneUpdateTime – lastChangeTime (NOTE 1) |
| ageOfContent | The time duration, in units of milliseconds, between the time the content is created in the scene by the Scene Manager and the time it is presented to the user. It can be computed as follows: actualDisplayTime – sceneUpdateTime (NOTE 1) |
| sceneUpdateDelay | The time duration, in units of milliseconds, spent by the Scene Manager to update the scene graph. It can be computed as follows: startToRenderAtTime – sceneUpdateTime |
| metadataDelay | The time duration, in units of milliseconds, between the time the split rendering metadata message is sent from the split rendering client and the time the split rendering server start to render using that metadata. It can be computed as follows: startToRenderAtTime – sendingAtTime |
| dataFrameDelay | The time duration, in units of milliseconds, spent to transmit the media rendered frame from the split rendering server to the split rendering client. It can be computed as follows: receptionTime – serverTransmitTime |
| NOTE 1: for the latency metrics computation, the timing information mentioned above need to be converted to a single time format (e.g., Wall clock time). | |

\* \* \* End of Changes \* \* \* \*