**Source**: Interdigital Finland Oy

**Title:** [MeCAR] User interaction QoE

**Agenda item:** 9.5

**Document for:** Discussion and Agreement

#### 1 Discussion

In use cases such as shared interactive immersive services, the user interaction is sent from a UE to a server. The server handles the user’s request to the immersive media scene (e.g., changing the context such as translation, rotation, and scaling or adding a new object in the scene). With the edge assisted UE type, the UE offloads the scene rendering to the Split Rendering Server, the Split Rendering Server rasterizes the XR viewport and does pre-rendering to generate a XR media which is encoded and delivered to the UE.

One important parameter to estimate the user quality of experience is the roundtrip interaction delay which is defined in the TR 26.928 [1] section 4.2.2. “Interaction Delays and Age of Content”.

We propose to detail the procedure with the different steps to measure the interaction delays and estimate the interactivity QoE. To measure these delays, we propose to introduce the sceneUpdateTime timestamp when the scene manager starts to process the user action.

#### 2 Proposed changes

--------------------------------------------- Begin change -------------------------------------------------------------------------------

1. 8.X User interaction

In use cases such as shared interactive immersive services, the user interaction is sent from a UE to a server. The server handles the user’s request to the immersive media scene (e.g., changing the context such as translation, rotation, and scaling or adding a new object in the scene). With the edge assisted UE type, the UE offloads the scene rendering to the Split Rendering Server, the server rasterizes the XR viewport and does pre-rendering to generate a XR media which is encoded and delivered to the UE.

1. 8.X.1 QoE timing information

In the context of interactive immersive services, one important parameter to estimate the user quality of experience is the *roundtrip interaction delay* which is defined in the TR 26.928[1]section 4.2.2. “Interaction Delays and Age of Content”.

The *roundtrip interaction delay* is defined as the sum of the *Age of Content* and the *User Interaction Delay.*

The *User interaction delay* is defined as the time duration between the moment at which a user action is initiated and the time such an action is taken into account by the content creation engine. It is impacted by the uplink latency of the wireless network.

The*Age of content* is defined as the time duration between the moment the content is created and the time it is presented to the user. It is impacted by the downlink latency of the wireless network.

The interactivity Quality of Experience (QoE) is highly dependent on the *roundtrip interaction delay*. Furthermore, the acceptable delay may differ according to the use case and the type of interaction. Several type of interactions with different roundtrip interaction delay thresholds may coexist in one application.

The Table X lists four categories of interaction/application defined in the TR 26.928 [1] with respect to roundtrip interaction delay:

|  |  |
| --- | --- |
| **Interaction/application Categories** | **Roundtrip interaction delay threshold** |
| **Ultra-Low-Latency** | ≤ 50 ms |
| **Low-Latency** | ≤ 100 ms |
| **Moderate latency** | ≤ 200 ms |
| **Non-critical latency** | > 200 ms |

Table X

The user interaction may be a single event which is utterly asynchronous from other data flows or may occur at different frequency.

1. 8.X.2 QoE measurement

For the Edge dependent MeCAR XR device type, the server’s split render function is in charge to pre-render the scene for the UE using the latest user pose, encodes the rendered frame and sends it back to the UE. The UE decodes the rendered frame, performs further post-processing like pose correction before presenting the frame to the user.

The procedures of the interactivity pipeline are detailed in the following Figure XYZ.



Figure XYZ: User action call flow

**Procedure:**

1. The raw user action is acquired from the XR runtime by the XR source management, including the *lastChangeState* timestamp of the last change to the state of this action.
2. The XR source management formats and gathers the raw user action into an action message. The action message is shared with the Media access Function (MAF).
3. The MAF sends the action message to the Scene Manager in the Split Rendering Server.
4. The action message is received by the Split Rendering Server and buffered before being handled during the next iteration of the update loop of the Scene manger. The Scene manager in the server processes the interaction task according to the actions in the action message from the UE and updates the scene. The Scene Manager records in the *timeInfo* metadata the *sceneUpdateTime* timestamp when it starts to process the action.
The scene manager may ignore the action according to the application policy: too many actions, actions too late or lower priority.
5. The Scene manager shares the updated scene with the renderer.
6. The scene is rendered using the predicted user pose.
7. The rendered media frame is shared with the Media Delivery Function.
8. The Split Rendering Server encodes the rendered media frame together with the associated time metadata. The encoded media frame is sent from the Split Rendering Server to the UE MAF with the associated time metadata.
9. The UE MAF decodes the media data. The rendered frame is then shared to the Presentation Engine and the XR runtime.
10. The XR runtime performs further processing such pose correction using the latest pose.
11. The rendered frame with the interaction response is displayed to the user at the actual display time (T2.actual).
12. 8.X.3 Interactivity delays and QoE

Using all the timestamps from the Split Rendering Server and the UE, the application can calculate the interaction delays:

* *User-interaction-delay = sceneUpdateTime - lastChangeState*
* *Age-of-content = T2.actual - sceneUpdateTime*
* *Roundtrip-interaction-delay = T2.actual - lastChangeState*

Those delays allow to estimate the interactivity QoE taking into account the category of interaction defined in the TR 26.928 [1] and listed in the Table X.

By using all the history of delay measurement, the Application can estimate the delays between the next user action and rendered frames.

* Action and timestamp information from the device:
	+ Action information: the user action information which are grouped into action sets. Each action has a unique identifier of the action
	+ *lastChangeTime*: the time when the user action is made. It corresponds to the lastChangeTime field in the action information defined as the timestamp of the last change to the state of the action.
* Action and timestamp information associated with rendered media frame from the Split Rendering Server:
	+ Action identifiers: The identifiers of the actions which are handled by the scene manager and rendered in the associated media frame
	+ *sceneUpdateTime*: the time when the Scene manager processes the interaction task according to the actions in the action message from the UE and updates the scene.

--------------------------------------------- End First change -------------------------------------------------------------------------

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

#### 3 Proposal

It is proposed to add the proposed changes in clause 2 to the sections .8.X of MeCAR PD v0.5.1.