Source: Samsung Electronics Co., Ltd.

**Title: ITT4RT: On viewport independent and viewport dependent delivery**

**Agenda Item: 11.5**

**Document for: Discussion and Agreement**

# **Introduction**

This contribution aims to trigger discussions on certain aspects of viewport independent and viewport dependent delivery in ITT4RT. So far, there has been many discussions on solutions related to the viewport-dependent delivery topic, including viewport margins, event based RTCP viewport mechanisms etc. Whilst these solutions (and their SDP signalling implementations) have been thoroughly discussed individually, the group should also consider the use of such solutions together as an end-to-end ITT4RT service.

1. **Discussion**

**Viewport-independent delivery / viewport-dependent delivery**

Viewport-dependent delivery may have different levels of complexity depending on the exact configuration, and certain configurations maybe not require constant (low latency) viewport feedback, or dynamically changing viewport margins. Additional media processing complexity for such mechanisms (either at the sender or network entity (MRF/MCU)) should not compromise the live requirements of the conversational service. Since OMAF solutions assume a set of pre-processed 360 video elementary streams (e.g. sub-picture region-wise packed multi-tracks in a file format) for streaming, it can be anticipated that the live conversational requirements of ITT4RT will further restrict the possibility of media processes that can be achieved.

It may be useful to categorise viewport-independent/dependent delivery into further configurations depending on media processing complexity and use case, such as:

1. Viewport independent delivery
   * Media sent by the ITT4RT-Tx client is pre-defined and cannot be selected
   * Delivery of receiver (ITT4RT-Rx client) viewport information is not required, and is only used by the receiver for rendering
2. Receiver selected viewport dependent delivery
   * The field-of-view of the media in multiple video streams is defined during session negotiation and is not changed (each video stream which is region packed may contain either the whole 360 video, or only a part of the 360 video)
   * The initial desired field-of-view (multiple video streams) to be received by the ITT4RT-Rx client is selected and defined as part of the session negotiation outcome, and can be changed using signalling or other means
   * Delivery of receiver viewport information is required only if the ITT4RT-Rx client does not have information on the set of streams with pre-defined FoV available, or if the ITT4RT-Rx client is unable to select the desired streams
3. Dynamic viewport driven viewport dependent delivery
   * Media sent by the ITT4RT-Tx client is dynamically processed such that it matches the viewport of the ITT4RT-Rx client
   * The receiver’s (ITT4RT-Rx client) viewport is sent to the sender ITT4RT-Tx client via “Viewport” RTCP feedback messages

Viewport independent processing (VIP) and viewport dependent processing (VDP) as previously discussed by the group represent the cases described by 1) and 3), respectively, whilst 2) is a viewport based delivery use case which can be achieved with less media processing complexity and less stringent latency requirements for feedbacks between the ITT4RT-Rx and ITT4RT-Tx clients.

Since 2) assumes the creation of a set of multiple video streams each with a defined field-of-view, it has the following characteristics:

* Less sensitive to latencies in delivery/receipt of RTCP feedback messages since media generation is not reliant on the viewport information contained in the same (“Viewport”)
* Well suited to scalability to support multiple users in a session since immersive 360-video is uni-directional in ITT4RT, and originates from the same source for all receiving users (the case described in 3) is less suited to scalability since it relies on user customised (viewport) encoded media, as well as the receipt of the “Viewport” RTCP feedback message)
* Less complexity compared to 3): depending on the complexity of the specific processing required for the video streams, MRF/MCU may not be required; in the case of 3), customised encoded media for multiple users would definitely require the use of an MRF/MCU

# **Proposal**

We propose to include the discussion text in section 2 into the next updated version of the PD document, and to further look into methods of supporting the end-to-end ITT4RT service as described in bullet point 2).