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
   * Sender generates media streams based on predetermined viewport orientations
   * Receiver selects a stream depending on its viewport information, and sender delivers the stream
   * The streams provided each contain the whole 360 video, but each with a different region of high quality such that streams are selected based on this region of high quality
   * The receiver only signals to the sender to select a different stream when its viewport information deviates from the region of high quality inside the current stream
3. Viewport report driven viewport dependent delivery with scalable media generation
   * Sender generates media based on predetermined viewport orientations or as tiled media.
   * Receiver reports the viewport information using RTCP viewport reporting and sender delivers the tiles/streams suitable for that orientation.
   * The tiles/viewport orientations provide a FOV large enough to not require frequent media stream updates from the sender when viewport orientation changes.
   * The RTCP reporting interval for viewport can be adjusted to be less frequent when head motion is minimal since the reporting requirement is less stringent.
4. Viewport report driven viewport dependent processing and 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

When both 2) and 3) assume the creation of a set of multiple video streams each covering the whole 360 video, with a pre-defined region-wise packing and high quality region:

* In 2), receiver knows the set of streams available on negotiation, and information (e.g. region-wise information) concerning the stream being received is already known
* In 3), since only sender knows the set of streams available, when a stream is pushed to the receiver, information concerning the stream must be signalled (through RTP header extensions or etc)
* In 2), sending of an RTP feedback message (to select the stream) is actively driven directly as a result from the viewport orientation
* In 3), sending of an RTP feedback message (the viewport) is passive (based on interval), where the interval can be adjusted according to viewport motion velocity

Comparing 2), 3) and 4)

* 2) is well suited to large scalability where viewport dependent processing flexibility is limited since all users receive one of a predefined set of streams
* 3) is well suited to cases where a certain flexibility for viewport dependent processing is available in the sender (or MRF) since the sender can decide what custom combination of tiles/viewport orientations is sent to each user
* 4) is well suited to peer-to-peer cases since full dynamic viewport dependent processing (dynamic encoding) can be achieved depending on the receiver’s viewport.

# **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).