**Source:** Nokia Corporation[[1]](#footnote-2)

**Title:** Proposed changes to draft CR on Viewport Dependent Processing

**Document for** Agreement

**Agenda item:** 12.5

# Introduction

This document presents changes related to viewport-dependent processing, to be incorporated to the informative section X.9 of draft CR (S4-210430) to TS 26.114.

The important changes are listed below:

* A new parameter *["VL"]* is added to the attribute *3gpp\_360video*, which indicates whether the delivered 360-degree video is to be rendered as viewport-locked. The parameter is particularly relevant to VDP scheme that delivers only the viewport region. Viewport-locked mode always renders the received 360-degree video to the center of the current viewport of the receiver UE. This is suitable for 2D rendering and requires no additional bitstream signalling and reverse rotation processes. Details are also added to the information section X.9.
* A new parameter *viewport\_size* is added to the attribute *3gpp\_360video*, which indicates the size of the viewport of the ITT4RT-Rx client.
* Section X.9 on viewport-dependent processing text is revised to be appropriate for inclusion in the specification.
* Text pertaining to establishing multiple 360-degree streams (Low Quality background and High Quality viewport) and bandwidth that was moved to the draft CR from the ITT4RT permanent document (PD) was removed due to insufficient specification of this process. This text is available in the PD and can be incorporated later.
* X.6.2 Main 360-degree video

X.6.2.1 General

A new SDP attribute 3gpp\_360video is defined with the following ABNF syntax:

3gpp\_360video = "a=3gpp\_360video:" [SP FOV] [SP FOV\_CENTER] [SP "Stereo"] [VDP]

VDP = "VDP" [SP Projection] [SP PPM] SP viewport\_ctrl SP viewport\_size

NOTE: If the SDP negotiations become too complex, defining profiles can be considered.

The semantics of the above attribute and parameters is provided below. Unsupported parameters of the 3gpp\_360video attribute may be ignored.

An ITT4RT client supporting the 3gpp\_360video attribute using viewport-independent processing or stereoscopic video for video shall support the following procedures:

* when sending an SDP offer, the ITT4RT client includes the 3gpp\_360video attribute in the media description for video in the SDP offer,
* when sending an SDP answer, the ITT4RT client includes the 3gpp\_360video attribute in the media description for video in the SDP answer if the 3gpp\_360video attribute was received in an SDP offer,
* after successful negotiation of the 3gpp\_360video attribute in the SDP, for the video streams based on the HEVC or AVC codec, the ITT4RT clients exchange an RTP-based video stream containing an HEVC or AVC bitstream with omnidirectional video specific SEI messages as defined in clause X.3.

An ITT4RT client supporting the 3gpp\_360video attribute supporting use of viewport-dependent processing (VDP) shall include the VDP parameter in the SDP offer and answer. Depending on the value indicated by the VDP parameter, the ITT4RT client shall further support the following procedures:

* the RTCP feedback (FB) message described in clause X.7.2 of type ‘Viewport’ to carry requested viewport information during the RTP streaming of media (signalled from the ITT4RT-Rx client to the ITT4RT-Tx client).

An ITT4RT client shall not include VDP parameter in the SDP answer if the SDP offer contains the 3gpp\_360video attribute without the VDP parameter.

An ITT4RT-Rx client that supports viewport-locked VDP shall render the received 360-degree content such that its center is aligned to the center of the current viewport of the ITT4RT-Rx UE. This is in contrast to normal rendering of VDP content, in which it is necessary to map the delivered content to the global coordinates (real capture position) of the 360-degree video, i.e, the normal rendering is sphere-locked.

Viewport-locked VDP is suitable for devices with 2D screens. ITT4RT-Tx clients may fall back to viewport-locked mode if they detect a 2D ITT4RT-Rx UE. ITT4RT-Tx clients that deliver viewport-locked VDP should only encode and deliver the viewport region as the region beyond the viewport will not be rendered at the ITT4RT-Rx client.

Further details on VDP can be found in section X.9.

X.6.2.2 Projection

An ITT4RT- client supporting the 3gpp\_360video attribute with VDP supporting projection may include the Projection parameter indicating the types of projection (e.g. ERP, CMP) it prefers (in the order of preference) in the SDP. An ITT4RT client may respond to an SDP offer with multiple options indicated in the Projection parameter with the agreed option. An ITT4RT-Tx client is not required to provide the preferred form of projection indicated by an ITT4RT-Rx client but may do so when possible.

The ABNF syntax is defined as follows:

Projection = “projection=ERP” / “projection=CMP”

X.6.2.3 Field Of View (FOV)

An ITT4RT-Tx client may support sending a limited 360-degree video.

* 1. An ITT4RT-Tx client supporting the 3gpp\_360video attribute capable of sending a limited 360-degree video shall include the parameter FOV in its SDP offer to indicate the cfov (*Capture FoV*) as the extent (range) of the 360-degree video with respect to the unit sphere. The range is expressed in unit of degrees with an *x* parameter for azimuth range and a *y* parameter for elevation range, sent as a comma-seperated tuple. In the absence of cfov, the default value of x and y are 360 and 180 degrees, respectively.
  2. An ITT4RT-Rx client supporting the 3gpp\_360video attribute capable that wants to receive a limited 360-degree video shall include the parameter FOV in its SDP offer/answer to indicate the pfov (*Preferred FoV)*, where *pfov* <= *cfov* in one or both the *x* and *y* dimensions when cfov is known. The *pfov* range is expressed in unit of degrees with an *x* parameter for azimuth range and a *y* parameter for elevation range, sent as a comma-seperated tuple. In the absence of pfov, it should be assumed to be the same as cfov.
  3. An ITT4RT-Tx client that has received an SDP offer from an ITT4RT-Rx client with the parameter FOV shall include in its SDP answer the parameter FOV to indicate the range of the 360-degree video it will provide. The value is the same as the FOV in the SDP offer or different based on the ITT4RT-Tx client capabilities.

An ITT4RT client supporting the 3gpp\_360video attribute with the FOV parameter may include the paramater FOV\_CENTER in the SDP. FOV\_CENTER is expressed as a comma-separated tuple (x,y), where x is the azimuth (in degrees) and y is the elevation (in degrees) with respect to the global coordinates such that the range defined FOV bypasses through the coordinates defined by FOV\_CENTER.The *imageattr* attribute indicates the resolution of the delivered content based on the *cfov* and *pfov* options.

The ABNF syntax is defined as follows

FOV = "fov=" 1\*(fovset)

fovset ="[x=" azimuthrange ",y=" elevationrange "]"

azimuthrange = azimuthvalue

elevationrange = elevationvalue

FOV\_CENTER = "fov=[x=" centerazimuth ",y=" centerelevation "]"

centerazimuth=azimuthvalue

centerelevation=elevationvalue

azimuthvalue = ["-"]1\*3DIGIT"."1\*3DIGIT

; An optional minus "-" that is

; followed by 1 to 3 other digits

; followed by "." to indicate fractional part

; followed by 1 to 3 other digits

elevationvalue = ["-"]2DIGIT"."3DIGIT

; An optional minus "-" that is followed by

; followed by 1 to 2 other digits

; followed by "." to indicate fractional part

; followed by 1 to 3 other digits

DIGIT = "1" / "2" / "3" / "4" / "5" / "6" / "7" / "8" / "9"

azimuthrange is a fractional value in the range of 0 to 360, inclusive

elevationrange is a fractional value in the range of 0 to 180, inclusive

centerazimuth is a fractional value in the range of -180 and 180, inclusive

centerelevation is a fractional value in the range of -90 and 90, inclusive

X.6.2.4 Picture Packing

An ITT4RT-Tx client may support sending the 360-degree video using tiled encoding, e.g., HEVC motion-constrained tiling. An ITT4RT-Tx client using the tiled encoding and supporting the 3gpp\_360video attribute shall include in its SDP offer the parameter PPM.

* A PPM value of 1 indicates mixed-quality tiled encoding is used
* A PPM value of 2 indicates mixed-resolution tiled encoding is used

An ITT4RT-Tx client sending low-quality background 360-degree video frame-packed with a high-quality viewport (possibly with margins) shall include in its SDP offer the parameter PPM.In this case the PPM is an ordered comma-seperated list of the following six fields:

* PPWHQ defines packed\_picture\_width of the high-quality region in pixels
* PPHHQ defines packed\_picture\_height of the high-quality region in pixels
* TRHQ defines transform operations applied on the high-quality region.
* PPWLQ defines packed\_picture\_width of the low-quality region in pixels
* PPHLQ defines packed\_picture\_height of the low-quality region in pixels
* TRLQ defines transform operations applied on the low-quality region

The transform operations have a value of 0-7 where the values correspond to the following operations:

|  |  |
| --- | --- |
| **Transform value** | **Transform operation** |
| 0 | no transform |
| 1 | mirrored horizontally |
| 2 | rotation by 180 degrees (counter-clockwise) |
| 3 | rotation by 180 degrees (counter-clockwise) before mirroring horizontally |
| 4 | rotation by 90 degrees (counter-clockwise) before mirroring horizontally |
| 5 | rotation by 90 degrees (counter-clockwise) |
| 6 | rotation by 270 degrees (counter-clockwise) before mirroring horizontally |
| 7 | rotation by 270 degrees (counter-clockwise) |

An ITT4RT-Rx client shall render the high-quality viewport region where these two regions are overlapping.

The ABNF syntax is defined as follows:

PPM = "ppm=" "1" / "2" / "["PPWHQ","PPHHQ","TRHQ","PPWLQ","PPHLQ","TRLQ"]"

Note: The size of the viewport and fov attributes define the size of projected regions. It should be considered if the two values should be included explicitly here

X.6.2.5 Viewport Control

An ITT4RT client that supports the 3gpp\_360video with VDP shall in its SDP offer include the parameter viewport\_ctrlwith one or more of the following control options:

* *device\_controlled* if ITT4RT-Tx client will provide VDP based on the requested viewport indicated by the RTCP feedback (FB) message type ‘Viewport’ sent by the corresponding ITT4RT-Rx client.
* *recommended\_viewport* if ITT4RT-Tx client will provide VDP with the help of a recommendation/prediction engine.
* *presenter\_viewport* if ITT4RT\_Tx will provide VDP based on the viewport of an ITT4RT-Rx client other than the one the SDP offer is being sent to.

Multiple options are provided as a comma-separated list. An ITT4RT client that receives an SDP offer with multiple viewport\_ctrl options may include its preferred viewport\_ctrl option in the SDP answer. If no options are given in the answer, the sender shall use the first option in the list. If the recommended\_viewport is successfully negotiated as viewport\_ctrl, the ITT4RT-Rx client should not use viewport prediction when sending the RTCP feedback (FB) message type ‘Viewport’ to avoid any conflicts with the prediction engine of the ITT4RT-Tx client. The ABNF syntax is defined as follows:

viewport\_ctrl = "viewport\_ctrl=" ("device\_controlled"/"recommended\_viewport"/"presenter\_viewport") \*2[(", device\_controlled" / ", recommended\_viewport" / ", presenter\_viewport")]

X.6.2.6 Overlay and 360-degree video

An ITT4RT client that sends an SDP message with at least one 360-degree video/audio and at least one overlay shall include in SDP the attribute *itt4rt\_group* before any media lines. The *itt4rt\_group* attribute is used to group 360-degree media and overlay media using the mid attribute as defined according to the ABNF below:

a = itt4rt\_group: <mid1> SP <mid2> SP <mid3> …

The list of mids in the itt4rt\_group shall include at least one mid associated with 360-degree media and at least one mid associated with an overlay as defined by the mid attribute in the corresponding media description.

An ITT4RT-Tx client and an ITT4RT-Rx client may negotiate the overlays that can be associated with the 360-degree video offered by the ITT4RT-Tx client using the itt4rt\_group attribute. An ITT4RT client shall indicate in an offer the overlays to be grouped with the 360-degree video using the itt4rt\_group attribute. The overlays that are acceptable shall be retained in the answer and the ones that are not acceptable shall be removed. An ITT4RT-Tx client may offer overlay configuration options using the 3gpp\_overlay attribute based on the list of media lines (i.e., potential overlay sources) provided in the itt4rt\_group attribute in an SDP offer initiated by an ITT4RT-Rx client. The 3gpp\_overlay attribute is offered in an SDP renegotiation.

The order of the media included in the itt4rt\_group indicates the synchronization source with the first media always being the synchronization anchor when synchronization is required.

NOTE: 2D video received from an ITT4RT-Rx clients may be offered as an overlay by the ITT4RT MRF to other ITT4RT-Rx clients. The ITT4RT MRF (acting as the ITT4RT-Tx client) would be the source of overlay media in this case.

X.6.2.7 Viewport Size

An ITT4RT client that includes the 3gpp\_360video with the VDP parameter shall also include in SDP the parameter viewport\_sizeto indicate the size of the device viewport using the azimuth and elevation ranges expressed in degrees. An ITT4RT-Tx client may include the viewport\_size of the ITT4RT-Rx client when this is known (e.g., in response to an SDP offer from an ITT4RT-Rx client) or include "viewport=0x0" and the value can be ignored by the ITT4RT-Rx client.

The ABNF syntax is given below:

viewport\_size = "viewport=" azimuthrange"x"elevationrange

where azimuthrange is a non-fractional value with the range 0-180 inclusive, and the elevationrange is a non-fractional value with the range 0-360 inclusive expressed. For example, "viewport=110x90".

Note that the viewport size defines the size of the viewport of the ITT4RT-Rx client UE. An ITT4RT-Tx client provides VDP based on this viewport size. The capture and preferred FOV, on the other hand, defines the range of the 360-degree content, which should be larger than the viewport size and can be negotiated even for viewport-independent processing.

X.9 Viewport-Dependent Processing (Informative)

Note: Informational text about VDP should be included in this section. Considerations for interoperability of ITT4RT clients using different VDP options need to be considered and normative text, if any, would be added to the sections above. The section provides a non-exhaustive list as guidance.

An ITT4RT-Tx client may offer Viewport Dependent Processing for delivering bandwidth-efficient 360-degree video to ITT4RT-Rx clients when both ends have successfully negotiated the required capabilities. There may be several ways in which the ITT4RT-Tx client may support VDP. This section lists some ways as informational with appropriate required signalling. This is a non-exhaustive list for guidance. Implementations may use other methods not defined here or a combination as long as the appropriate capabilities and required signalling is supported by ITT4RT clients.

## X.9.1 360-degree Video Optimized for the Viewport

An ITT4RT-Tx client may deliver the full 360-degree video as a single encoded stream with higher quality in the viewport region. This may be implemented by encoding parameters which provide higher quality in the viewport region compared to the other regions. The bitrate savings are limited when the full 360-degree video is delivered. However, a delay in viewport update does not cause disruption in the viewing experience in terms of content flow, but just a temporary lowered quality of the visible content.

The case described here is to provide an omnidirectional video to receiver UEs without the need for unpacking operation(s). It should be noted that region-wise packing may still be used with such encoding, if desired. Tiling, which may lead to lowered encoding efficiency, is not required in this case but multi-quality tiles may still be used without need for packing. For the sake of scalability, an MRF/MCU delivering content to multiple end users may receive viewport independent 360-video from the content source, and encode multiple versions of the content optimized for different viewing orientations, and deliver the version most suitable to the current viewport of each user.

The full processing pipeline consists of the ITT4RT-Tx client receiving an image from a capture device and stitching it, if not pre-stitched. It then determines the viewport of the receiver based on RTCP viewport feedback. Once the viewport has been determined, the video is encoded with higher quality in the viewport region, packetized and delivered to the ITT4RT-Rx client. The ITT4RT-Rx client decodes and renders the received video. No additional signalling is needed.

X.9.2 Tiled Video with Multiple Quality Levels

An ITT4RT-Tx client may use tiled encoding, e.g., HEVC motion-constrained tiling, to create a tiled low-quality version of the 360-degree video and a tiled high-quality version of the 360-degree video (note that more than two quality levels for the encodings may be created). The delivered stream consists of high-quality tiles for the viewport region and low-quality tiles for the non-viewport region. When the low-quality and high-quality tiles are encoded using low fidelity and high fidelity, we refer to this as the mixed-quality tiled encoding approach. Whereas if the low-quality and high-quality tiles are created using lower resolution and higher resolution respectively, we refer to this as the mixed-resolution tiled encoding approach.

The main advantage of tiled encoding as described in this section is scalability, making it more suitable for implementation in an MRF. For example, a high-quality viewport-independent 360-degree video is delivered to an MRF, which re-encodes and delivers viewport-dependent 360-degree video to multiple ITT4RT-Rx clients The MRF does not have to produce content for each receiver individually, but the operations are limited to picking the right high quality tiles and low-quality tiles from the two versions of the video, and “assemble it” for each ITT4RT-Rx client based on its current viewport. However, tiled video has lower coding efficiency as inter-picture prediction is limited to each tile. Furthermore, the storage and processing requirements for encoding tiled video at multiple qualities may be unsuitable for smaller conference sizes and ITT4RT-Tx clients in terminal.

All tiles can be packaged and delivered in a single RTP stream and can be decoded using a single decoder when all tiles are decoded together. In case of mixed-resolution, packing information needs to be signaled to the receiver, which uses it for rendering after decoding. The SEI messages for region-wise packing (RWP) can be used to carry the information about packed picture mapping to assist the receiver in the understanding of the high-quality and low-quality areas. Note that multiple streams and multiple decoders may be used for the tiled case, but may increase complexity.

For the case of tiled video with multiple quality levels, the encoding process is the same regardless of the viewport of the ITT4RT-Rx client. Once encoding is done, the ITT4RT-Tx client can determine the viewport and select the tile qualities based on the viewport region. The tiles are packed and delivered to the ITT4RT-Rx client, which decodes and renders. The stream would carry information about RWP (using SEI) to assist the receiver in unpacking. The unpacking is part of the rendering process and may require upsampling tiles and arranging the upsampled tiles into the projected format in the mixed-resolution case. The packing/unpacking step and signalling packed picture information in the stream is not needed for the mixed-quality case.

X.9.3 Viewport-only Delivery

For maximim bandwidth savings, the ITT4RT-Tx client may deliver only an extracted high-quality region corresponding to the viewport of the ITT4RT-Rx client. If the viewport region (with or without a viewport margin) is extracted from a projected picture (e.g., ERP), the resolution would change depending on where the viewport is located on the picture. To prevent changing resolution, the ITT4RT-Tx client may rotate the sphere to re-orient the selected viewport to the center of the projected picture. The delivered video may be rendered in one of two ways, which affect the signalling requirements i) sphere-locked, where the ITT4RT-Rx client reverses rotation of the received image before rendering, and ii) viewport-locked, where the ITT4RT-Rx client always renders the received image to the center of the viewport. Note that sphere-locked operation implies that it is necessary to map the delivered content to the global coordinate (real capture position) of the stitched 360-degree video. This is the default behaviour when VDP is used without the VL parameter regardless of the VDP method.

For this case, the ITT4RT-Tx client receives an image from a capture device/source and stitches it, if not pre-stitched. It then determines the viewport region (with or without a viewport margin) of the ITT4RT-Rx client based on RTCP viewport feedback. Once the viewport region has been determined, the center of the viewport region is rotated to the center of the projected picture and encoded. Depending on whether sphere-locked or viewport-locked operation is used, the following apply:

* For sphere-locked operation, the ITT4RT-Tx client must signal the rotation information to the ITT4RT-Rx client using the rotation SEI message. The ITT4RT-Rx client then reverses the rotation before rendering.
* For viewport-locked operation, the ITT4RT-Tx client does not need to signal the rotation information in the bitstream. The ITT4RT-Rx client renders the received video such that the center of the image is aligned to the center of the current viewport.

In both cases, the ITT4RT-Rx client always receives a constant sized, constant resolution image for the viewport region. The viewport-locked solution may be useful for any 2D display devices, as a delayed viewport update (at least 1 RTT) does not significantly lower the user experience. In case of ultra low-latency networks, the solution can be used for HMDs, but may cause motion sickness if latency increases and, hence, must be used carefully.

X.9.4 HQ Viewport with LQ background

A viewport-only solution, like the one described in X.9.6, can be combined with a low-quality (LQ) viewport-independent 360-degree video as background to deliver a more continuous viewing experience in case of head motion. The LQ viewport-independent 360-degree video is frame packed with the viewport region (with or without margin) at a higher quality (HQ) and delivered as a single stream as shown in Figure X.9.6. Since the size and shape of the pictures remain same, the packing information is signaled in SDP, using the Packed Picture Mapping (PPM) parameter as defined in X.6.2.4. The ITT4RT-Rx client would render the HQ viewport region [on top of] the LQ background. Further details on the signalling and pipeline of a single stream case follow.

A high-quality viewport is encoded in the same way as described in x.9.3 for HQ viewport-only delivery. In addition, a LQ 360-degree viewport-independent version of the video is encoded. Both streams are packed together, packetized and delivered to the ITT4RT-Rx client. The ITT4RT-Rx client unpacks, decodes and reverse rotates the HQ viewport to the right coordinates. The viewport region is rendered on top of the low-quality background.

X.9.6 Viewport Margins

Editor’s Note: Signalling the extent of margins is FFS. The section below is informative about the use of viewport margins in viewport-dependent delivery.

When VDP is used, a change in viewport, e.g., due to head motion at the ITT4RT-Rx client in terminal, may require an update in the viewport region. This change is triggered by an RTCP feedback with the new viewport information. It may take up to at least one Round Trip Time (RTT) or more for the viewport to update, resulting in a motion-to-high-quality delay. Motion-to-high-quality delay is the amount of time it takes for the new viewport to reach comparable quality to the early viewport after head motion. An ITT4RT-Tx client that support VDP may use viewport margins to minimize this delay and also to reduce the need for frequent viewport updates.

Viewport margins can be extended on all or some sides of the viewport and may be at the same quality (or resolution) as the viewport or at a quality (or resolution) lower than the viewport but higher than the background. Viewport margins may be extended around the viewport evenly or unevenly. Some example scenarios where viewport margins may be used to improve playout are listed below:

* Equally extended margins (symmetric) around the viewport in all directions may be used to decrease the motion to high-quality delay. The margins may be gradually extended farther by probing the network and reduced when the network is congested. In this scenario, the use of margins is akin to sending a larger viewport.
* Margins may be unevenly extended around the viewport (directional) with larger margins in the direction the user’s head is turning. In the absence of head motion, the margins may return to being equally extended all around the viewport. In this case, RTCP viewport feedback is used to decide the distribution of margins. See Figure x.9.8.
* Margins may be unevenly extended around the viewport with larger margins in the direction of the predicted head motion, e.g., based on audio input, motion tracking or other application level functions.

|  |  |  |  |
| --- | --- | --- | --- |
| a.head turning right | b. head turning left | c. head turning up | d. head turning down |

**Figure x.9.8: Uneven extension of margins based on the user head motion. Similar uneven extensions may be used based on other application-level parameters. Symmetric margins extend evenly top, bottom and left, right of the viewport.**

\*\*\*\*End changes\*\*\*\*

# Proposal

The proposal is to include the proposed changes to the draft CR on viewport-dependent delivery, Tdoc S4-210430.

1. Contact: Igor Curcio, Saba Ahsan, Nokia, Finland. Emails: íigor.curcio, saba.ahsaný@nokia.com. [↑](#footnote-ref-2)