|  |  |
| --- | --- |
| 3GPP TR 26.862 V0.2.0 (2022-02) | |
| Technical Report | |
| 3rd Generation Partnership Project;  Technical Specification Group Services and System Aspects  Immersive Teleconferencing and Telepresence for Remote Terminals (ITT4RT) Use Cases, Requirements and Potential Solutions  (Release 17) | |
|  | |
|  |  |
|  | |
| The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP. The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented. This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification. Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices. | |

|  |
| --- |
|  |
| ***3GPP***  Postal address  3GPP support office address  650 Route des Lucioles - Sophia Antipolis  Valbonne - FRANCE  Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16  Internet  http://www.3gpp.org |
| ***Copyright Notification***  No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media.  © 2022, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).  All rights reserved.  UMTS™ is a Trade Mark of ETSI registered for the benefit of its members  3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  GSM® and the GSM logo are registered and owned by the GSM Association |

Contents

Foreword 4

1 Scope 5

2 References 5

3 Definitions of terms, symbols and abbreviations 6

3.1 Terms 6

3.2 Symbols 7

3.3 Abbreviations 8

4 Use Cases 8

4.1 Main ITT4RT Use Case 8

4.2 Use Case Extension 1: Viewport sharing among remote participants 12

4.3 Use Case Extension 2: Viewport-dependent stream for display device 13

4.4 Use Case Extension 3: Viewport sharing for second display device 15

4.5 Use Case Extension 4: Separate stream for presentation content 17

4.6 Use Case Extension 5: Applications to 3GPP FLUS 17

4.7 Use Case Extension 6: Multiple Overlays 18

4.7.1 Scenario 1 18

4.7.2 Scenario 2 19

4.8 Use Case Extension 7: Audio mixing of multiple streams in ITT4RT 19

4.8.1 Use Case 19

5 Requirements 20

6 Architecture 23

6.1 General 23

6.2 Potential architecture extensions 24

6.2.1 Network-based Media Processing 24

6.2.1.1 Introduction 24

6.2.1.2 Example Architecture 24

7 Potential Solutions 25

7.1 Frame packing of overlays 25

7.1.1 SDP Signalling and Examples 25

7.1.2 Bitstream Signalling 28

7.2 Conditional overlays 30

7.2.1 Introduction 30

7.2.2 Potential solution using RTCP 30

7.2.3 Potential solution using SDP 31

8 Selected solutions in TS 26.114 31

Annex A (informative): Change history 32

# Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

This Technical Report collects information related to enable “Immersive Teleconferencing and Telepresence for Remote Terminals” (ITT4RT) in 3GPP MTSI. ITT4RT is expected to enable scenarios with two-way audio and one-way immersive video, e.g., a remote single user wearing an HMD participates in a conference will send audio and optionally 2D video (e.g., of a presentation, screen sharing and/or a capture of the user itself) but receives stereo or immersive voice/audio and immersive video captured by an omnidirectional camera in a conference room connected to a fixed network.

This Technical Report serves in addition to the specification in TS 26.114 or TS 26.223 to capture a broader scope of technical solution for ITT4RT. One the one hand this is to present more details while keeping the TS documents lean and on the other hand to serve as a basis for further work.

In the present document, modal verbs have the following meanings:

**shall** indicates a mandatory requirement to do something

**shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

**should** indicates a recommendation to do something

**should not** indicates a recommendation not to do something

**may** indicates permission to do something

**need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

**can** indicates that something is possible

**cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

**will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

**might not** indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

**is** (or any other verb in the indicative mood) indicates a statement of fact

**is not** (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

# 1 Scope

The present document collects information regarding the “support of Immersive Teleconferencing and Telepresence for Remote Terminals” (ITT4RT) in 3GPP MTSI. The primary scope of the present document is the documentation of the following aspects:

* Core use cases and requirements for ITT4RT
* Architectures for ITT4RT
* Potential Solutions for ITT4RT not specified in TS 26.114 or TS 26.223
* Example Signalling flows for ITT4RT
* Selected Solutions as pointer to the TS 26.114 or TS 26.223

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 26.071: "Mandatory Speech Codec speech processing functions; AMR Speech CODEC; General description".

[3] 3GPP TS 26.190: "Speech codec speech processing functions; Adaptive Multi-Rate - Wideband (AMR-WB) speech codec; Transcoding functions".

[4] 3GPP TS 26.441: "Codec for Enhanced Voice Services (EVS); General Overview".

[5] 3GPP TS 26.171: "Speech codec speech processing functions; Adaptive Multi-Rate - Wideband (AMR-WB) speech codec; General description".

[6] 3GPP TS 26.118: "3GPP Virtual reality profiles for streaming applications".

[7] 3GPP TS 26.114: "IP Multimedia Subsystem (IMS); Multimedia telephony; Media handling and interaction".

[8] 3GPP TS 26.223: "Telepresence using the IP Multimedia Subsystem (IMS); Media handling and interaction".

[9] IETF RFC 7728 (2016): "RTP Stream Pause and Resume".

[10] 3GPP TS 26.238: "Uplink streaming".

[11] IETF Internet Draft, draft-ietf-rtcweb-data-channel-13 (2015): "WebRTC Data Channels" (WORK IN PROGRESS)

[12] ISO/IEC 23008-12: Information technology – MPEG systems technologies – Part 12: Image File Format

[13] ISO/IEC 23090-8: " Information technology -- Coded representation of immersive media -- Part 8: Network-based media processing".

# 3 Definitions of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the terms given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

**360-degree video:** A real-world visual scene captured by a set of cameras or a camera device with multiple lenses and sensors covering the sphere in all directions around the centre point of the camera set or camera device. The term 360-degree video may be used to include also limited 360-degree video.

**Limited 360-degree video:** A 360-degree video in which the visual scene does not cover the entire sphere around the center point of the camera set or camera device but only a part of it. A limited 360-degree video may be limited i) in the horizontal field to less than 360 degrees, or ii) in the vertical field to less than 180 degrees or iii) in both the vertical and horizontal fields.

**AMR, AMR-NB:** Both names refer to the AMR codec (3GPP TS 26.071 [2]) and are used interchangeably in this specification.

**Bitstream:** A bitstream that conforms to a video or audio encoding format.

**bitstream**: A sequence of bits that forms the representation of one or more coded video or audio sequences.

**CHEM:** The Coverage and Handoff Enhancements using Multimedia error robustness feature.

**Codec mode:** Used for the AMR and AMR-WB codecs to identify one specific bitrate. For example, AMR includes 8 codec modes (excluding SID), each of different bitrate.

**Constrained terminal:** UE that is (i) operating in radio access capability category series "M" capable of supporting conversational services, and/or (ii) a wearable device which is constrained in size, weight, or power consumption (e.g., connected watches), excluding smartphones and feature phones.

**DCMTSI client:** A data channel capable MTSI client supporting data channel media as defined in clause 6.2.10.

**DCMTSI client in terminal:** A DCMTSI client that is implemented in a terminal or UE. The term "DCMTSI client in terminal" is used in this document when entities such as MRFP, MRFC or media gateways are excluded.

**Dual-mono:** A variant of 2-channel stereo encoding where two instances of a mono codec are used to encode a 2-channel stereo signal.

**Evolved UTRAN:** Evolved UTRAN is an evolution of the 3G UMTS radio-access network towards a high-data-rate, low-latency and packet-optimized radio-access network.

**EVS codec:** The EVS codec includes two operational modes: EVS Primary operational mode (‘EVS Primary mode’) and EVS AMR-WB Inter-Operable (‘EVS AMR-WB IO mode’). When using EVS AMR-WB IO mode the speech frames are bitstream interoperable with the AMR-WB codec [3]. Frames generated by an EVS AMR-WB IO mode encoder can be decoded by an AMR-WB decoder, without the need for transcoding. Likewise, frames generated by an AMR-WB encoder can be decoded by an EVS AMR-WB IO mode decoder, without the need for transcoding.

**EVS Primary mode:** Includes 11 bit-rates for fixed-rate or multi-rate operation; 1 average bit-rate for variable bit-rate operation; and 1 bit-rate for SID (3GPP TS 26.441 [4]). The EVS Primary can encode narrowband, wideband, super-wideband and fullband signals. None of these bit-rates are interoperable with the AMR-WB codec.

**EVS AMR-WB IO mode:** Includes 9 codec modes and SID. All are bitstream interoperable with the AMR-WB codec (3GPP TS 26.171 ‎‎[5]).

**Field of View**: The extent of visible area expressed with vertical and horizontal angles, in degrees in the 3GPP 3DOF reference system as defined in TS 26.118 [6].

**Fisheye Video**: Video captured by a wide-angle camera lens that usually captures an approximately hemispherical field of view and projects it as a circular image.

**Frame Loss Rate (FLR):** The percentage of speech frames not delivered to the decoder. FLR includes speech frames that are not received in time to be used for decoding.

**ITT4RT client:** MTSI client supporting the Immersive Teleconferencing and Telepresence for Remote Terminals (ITT4RT) feature, as defined in Annex Y.

**ITT4RT-Tx client:** ITT4RT client only capable of sending immersive video.

**ITT4RT-Rx client:** ITT4RT client only capable of receiving immersive video

**ITT4RT MRF:** An ITT4RT client implemented by functionality included in the MRFC and the MRFP.

**ITT4RT client in terminal:** An ITT4RT client that is implemented in a terminal or UE. The term "ITT4RT client in terminal" is used in this document when entities such as ITT4RT MRF is excluded.

**Mode-set:** Used for the AMR and AMR-WB codecs to identify the codec modes that can be used in a session. A mode-set can include one or more codec modes.

**MSMTSI client:** A multi-stream capable MTSI client supporting multiple streams as defined in Annex S. An MTSI client may support multiple streams, even of the same media type, without being an MSMTSI client. Such an MTSI client may, for example, add a second video to an ongoing video telephony session as shown in Annex A.11. In that case, the MTSI client is an MSMTSI client only if it is fully compliant with Annex S.

**MSMTSI MRF:** An MSMTSI client implemented by functionality included in the MRFC and the MRFP.

**MSMTSI client in terminal:** An MSMTSI client that is implemented in a terminal or UE. The term "MSMTSI client in terminal" is used in this document when entities such as MRFP, MRFC or media gateways are excluded.

**MTSI client:** A function in a terminal or in a network entity (e.g., a MRFP) that supports MTSI.

**MTSI client in terminal:** An MTSI client that is implemented in a terminal or UE. The term "MTSI client in terminal" is used in this document when entities such as MRFP, MRFC or media gateways are excluded.

**MTSI media gateway (or MTSI MGW):** A media gateway that provides interworking between an MTSI client and a non MTSI client, e.g., a CS UE. The term MTSI media gateway is used in a broad sense, as it is outside the scope of the current specification to make the distinction whether certain functionality should be implemented in the MGW or in the MGCF.

**Omnidirectional media:** Media such as image or video and its associated audio that enable rendering according to the user's viewing orientation, if consumed with a head-mounted device, or according to user's desired viewport, otherwise, as if the user was in the spot where and when the media was captured.

**Operational mode:** Used for the EVS codec to distinguish between EVS Primary mode and EVS AMR-WB IO mode.

**Overlay:** A piece of visual media, rendered over omnidirectional video or image, or a viewport.

**Pose:** Position and rotation information associated to a viewport.

**Projected picture:** Picture that has a representation format specified by an omnidirectional video projection format.

**Projection:** Inverse of the process by which the samples of a projected picture are mapped to a set of positions identified by a set of azimuth and elevation coordinates on a unit sphere.

**Simulcast:** Simultaneously sending different encoded representations (simulcast formats) of a single media source (e.g., originating from a single microphone or camera) in different simulcast streams.

**Simulcast format:** The encoded format used by a single simulcast stream, typically represented by an SDP format and all SDP attributes that apply to that particular SDP format, indicated in RTP by the RTP header payload type field.

**Simulcast stream:** The RTP stream carrying a single simulcast format in a simulcast.

**Viewport**: Region of omnidirectional image or video suitable for display and viewing by the user.

## 3.2 Symbols

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

**ITT4RT** Immersive Teleconferencing and Telepresence for Remote Terminals

**AR** Augmented Reality

**DTLS** Datagram Transport Layer Security

**FLUS** Framework for Live Uplink Streaming

**GSM** Global System for Mobile Communications

**HEIF** High Efficiency Image File Format

**HEVC** High Efficiency Video Coding

**HMD** Head-Mounted Display

**IMS** IP Multimedia Subsystem

**IP**  Internet Protocol

**ITT4RT** Immersive Teleconferencing and Telepresence for Remote Terminals

**LTE** Long-Term Evolution

**MCU** Multipoint Control Unit

**MEF** Maintenance Entity Function

**MPEG** Moving Picture Experts Group

**MRF / MRFC** Multimedia Resource Function Controller

**MSMTSI** Multi-Stream Multimedia Telephony Service for IMS

**MTSI** Multimedia Telephony Service for IMS

**NAL** Network Abstraction Layer

**OMAF** Omnidirectional Media Format

**RTP / RTCP** Real-Time Transport (Control) Protocol

**SCTP** Stream Control Transmission Protocol

**SDP** Session Description Protocol

**SEI** Supplemental Enhancement Information message

**SIP** Session Initiation Protocol

**SMS** Short Message Service

**UE** User Equipment

**UMTS** Universal Mobile Telecommunications System

**VR** Virtual Reality

# 4 Use Cases

## 4.1 Main ITT4RT Use Case

A group of colleagues is having a meeting in conference room A (see Figure 1). The room consists of a conference table (for physically present participants), a 360-degree camera[[1]](#footnote-1), and a view screen. Two of their colleagues, Bonnie (B) and Clyde (C) are travelling and join the meeting through a conference call.

* Participants in conference room A use the screen to display a shared presentation and/or video streams coming from Bonnie and Clyde.
* Bonnie joins the conference from her home using a Head Mounted Display (HMD) and a camera that captures her video. She has a 360-degree view of the conference room.
* Clyde joins the conference from the airport using his mobile phone. He/she also has a 360-degree view of the conference room on his mobile screen and uses his mobile camera for capturing his own video.

Both Bonnie and Clyde can see the screen in the conference room as part of the 360-degree video. They also have the option to bring into focus any of the incoming video streams (a presentation or the other remote participant’s camera feed) using their own display devices. The manner in which this focused stream is displayed is a function of their display device and is not covered in this use case.

Within the 3GPP MTSI TS 26.114 [7] and Telepresence TS 26.223 [8] specifications, the above use case can be realized in two possible configurations, which are explained below. The participants are referred to as A, B and C from here onwards.

In the first scenario, shown in Figure 2.1, the call is set up without the support of any media-aware network elements. Both remote participants, B and C, send information about their viewport orientation to A, which in turn sends them a viewport-dependent video stream from the omnidirectional camera.

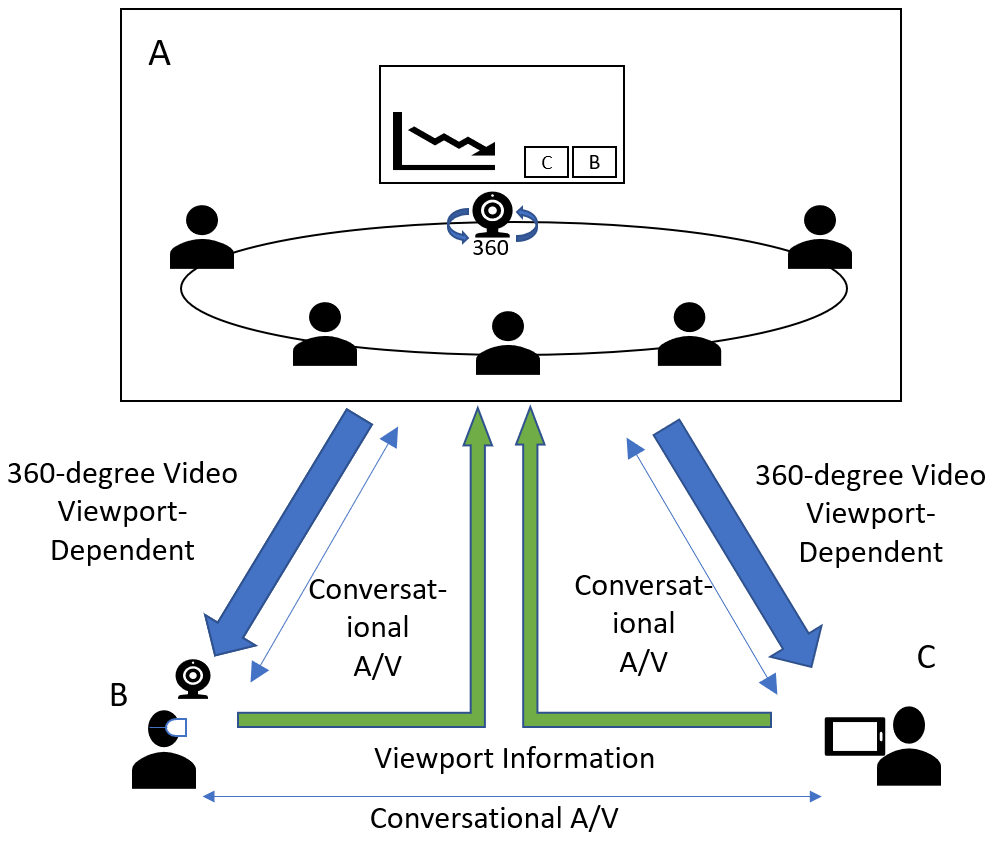


Figure 4.1.1 - 360-degree conference call

In the second scenario, the call is setup using a network function, which may be performed by either a Media Resource Function (MRF) [7] or a Media Control Unit (MCU) [8]. In this case, the MRF/MCU receives a viewport-independent stream from A. Both B and C, send viewport orientation information to the MRF/MCU and receive viewport-dependent streams from it. Figure 4.1.1 illustrates the scenario. The A/V channel for conversational non-immersive content also flows through the MRF/MCU in the figure.

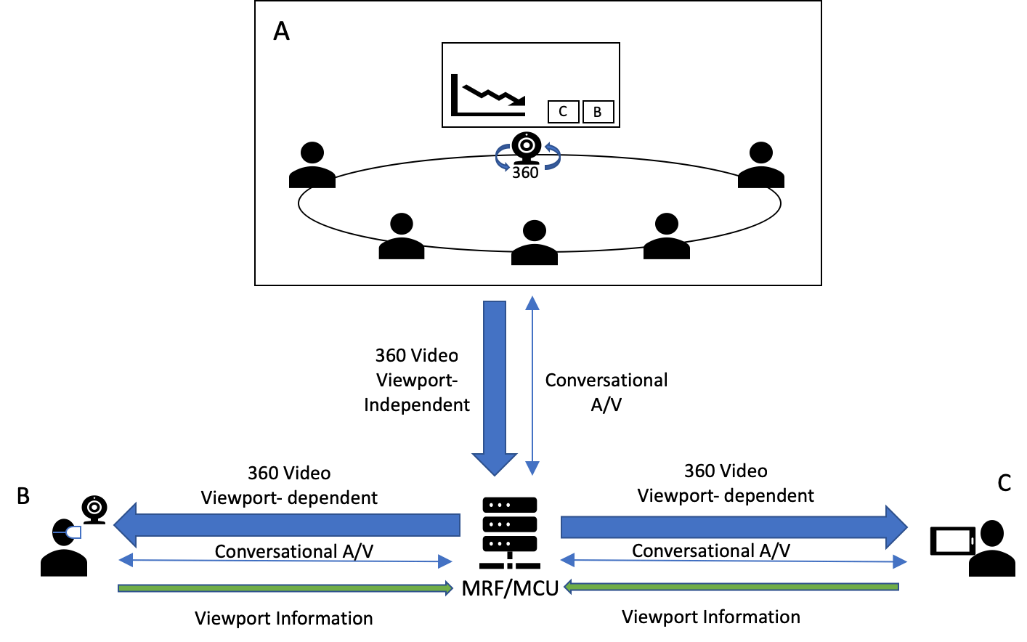


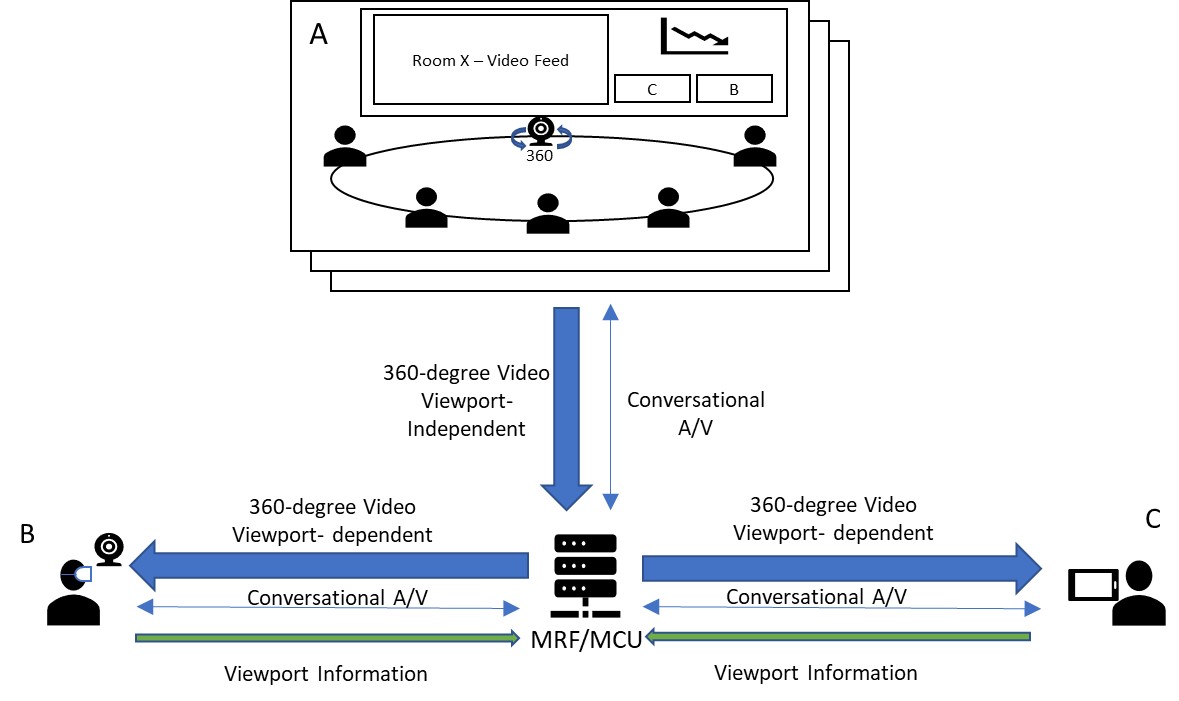
Figure 4.1.2 - A 360-degree conference call via MRF/MCU

The use case aims to enable immersive experience for remote terminals joining teleconferencing and telepresence sessions, with two-way audio and one-way immersive video, e.g., a remote single user wearing an HMD participates to a conference will send audio and optionally 2D video (e.g., of a presentation, screen sharing and/or a capture of the user itself), but receives stereo or immersive voice/audio and immersive video captured by an omnidirectional camera in a conference room connected to a fixed network.

Private side communication is also expected to be enabled as part of this use case. For instance, two users attending the conference may wish to talk privately and do not want to be heard by others. In this case, the audio information exchanged between these two users should not be transmitted to others, and the content of their conversation should be protected and only be fully rendered on their devices. Others may know that these users are interacting but would not be able to hear the specific content.

A special variation of this use case is when the 360 camera capture occurs not in a conference room but on a user device.

In the third scenario, multiple conference rooms are sending 360-degree video to an MRF/MCU. The rooms may choose to receive 2D video streams from other participants including one of the other rooms, which is displayed on the screen in the room. A pictorial representation is shown in Figure 4.1.3.



X

Figure 4.1.3: Multiple rooms with 360-degree video

Furthermore,

* The remote (single) users can choose to view any one or none (e.g., when only viewing the screenshare) of the available 360-degree videos from the multiple rooms. Switching from one room to another may be triggered manually, or using other mechanisms, such as, viewing direction or dominant speaker. However, for the case of the dominant speaker, it is important to consider effects of motion sickness while switching from one video to another as the user might not be sufficiently prepared for the switch.
* The MRF/MCU may signal to pause the receiving 360-degree video from any of the rooms that do not currently have any active viewers. The streams can be resumed if and when there are viewers [9].
* The presentation/screenshare stream is distributed to the single users and the rooms as a separate stream and can be identified using an explicit “a=content:slides” SDP field [7]. The single users may view the stream as an overlay on top of the 360-degree video.
* If the area of interest in the 360-degree video is limited (the preferred FoV is defined and negotiated between the participants) or the motion outside the FoV is limited, the background area (outside the viewport) can be transmitted as a still image to maintain continuity in the 360-degree view. The still-image may be updated at regular intervals or as needed.
* A still-image background area can also be used for placing overlays and avoiding unnecessary bandwidth utilization. In the scenario in the figure, this background area would be the wall with the screen and the presentation overlay may be placed on top of the screen for a better visual experience.
* Overlays may also be used for other 2D streams. These include videos from the single users and 2D video streams from the other rooms. Spatial audio should be associated with overlays when placed in the 360-degree view, i.e., the sound should appear to originate from the placement of the overlay, not necessarily associated with the elements within the video.
* MRF/MCU based media processing may be used for creating 360-degree views combining multiple input sources, e.g., 360-degree video with limited FoV, still image for background, overlays of screenshare and other 2D videos, etc., when the device capabilities are limited.

## 4.2 Use Case Extension 1: Viewport sharing among remote participants

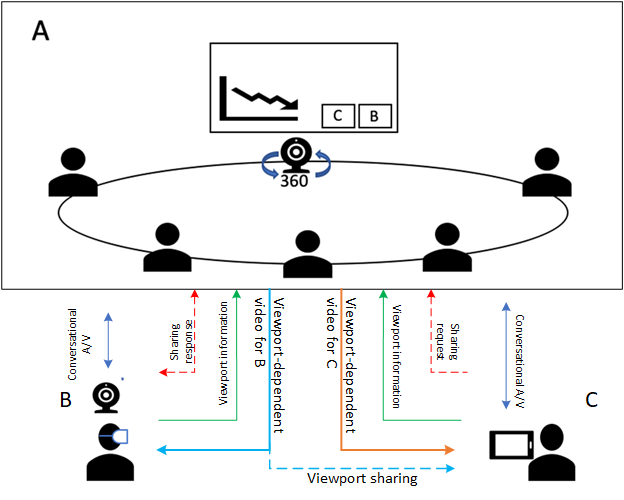


Figure 4.2.1 Viewport sharing between remote participants w/o MRF/MCU

Figure 4.2.1 illustrates viewport sharing capability added to the above use case. When B is presenting to the conference or B is communicating with C, C may be interested in B’s focus or her 360-degree viewport, especially when B is interacting with anything or anyone in room A. In such case, C would request to follow B’s viewport. Upon permission from B, C may follow B’s viewport on his own display device regardless of C’s orientation. In this scenario, room A may multicast (i.e., send the same information using unicast to different receivers as in usual MTSI/Telepresence) viewport dependent stream with embeddedB’s viewport metadata to both B and C. C’s device will follow embedded viewport metadata and playback the same viewport presented to B.

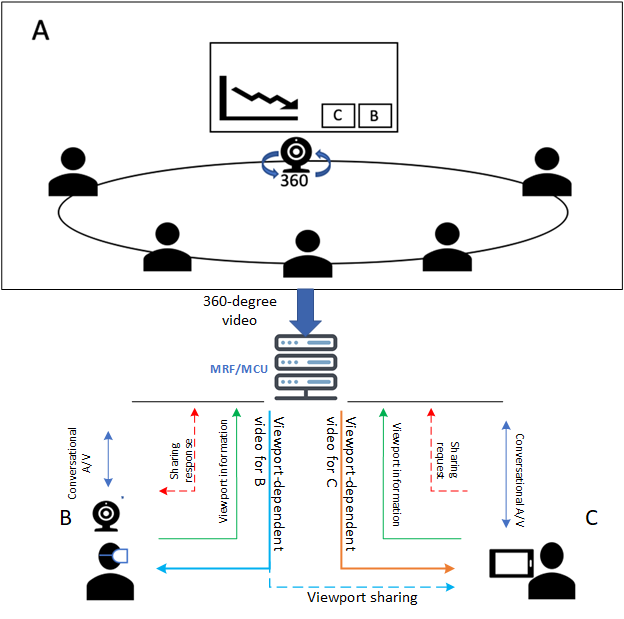


Figure 4.2.2 Viewport sharing between remote participants w/ MRF/MCU

Figure 4.2.2 shows the second scenario where the call is setup using a MRF or MCU. The MRF/MCU may receive C’s viewport sharing request and check B’s permission for such request. Once B’s permission is confirmed, MRF/MCU may forward the viewport dependent stream with B’s viewport metadata embedded to C. C’s device will follow embedded viewport metadata and playback the same viewport presented to B.

Viewport sharing feature may require capability of party A or MRF/MCU to forward B’s viewport-dependent video stream to C after request/response signalling exchange, and embed B’s viewport metadata into the stream.

## 4.3 Use Case Extension 2: Viewport-dependent stream for display device

Participants in conference room A may have their own display device (HMD, AR glass or mobile phone) to receive a 360-degree video stream (viewport independent or dependent stream) from room A, or conversational video from remote participant B or C. For a large conferencing room, some participants might not sit close to the view screen or other participants he or she would like to communicate, a display device may offer high quality view of room A regardless where the participant sits. A viewport-independent or viewport dependent 360-video stream may send to participant’s display device.

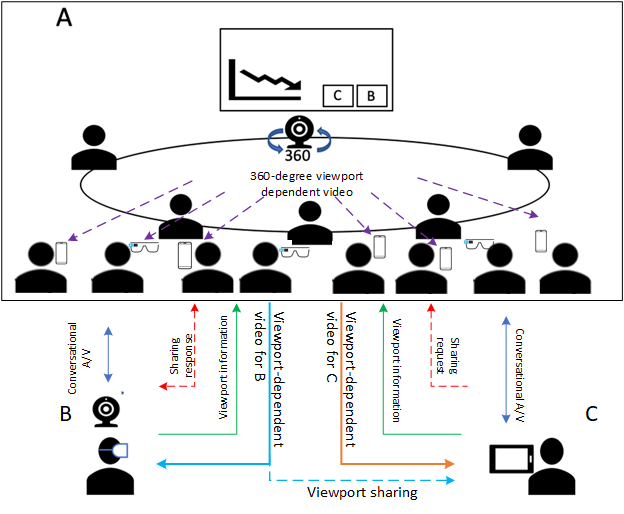


Figure 4.3.1 Proposed viewport sharing use case w/o MRF/MCU

Figure 4.3.1 shows the first scenario, the call is setup without the support of a MRF or MCU. B and C send viewing orientation information to A and receive corresponding viewport-dependent streams from the omnidirectional camera. For the participants with their own display device, the display device may receive 360-degree viewport-independent video from omnidirectional camera, or viewport-dependent video stream by sending orientation information to A.

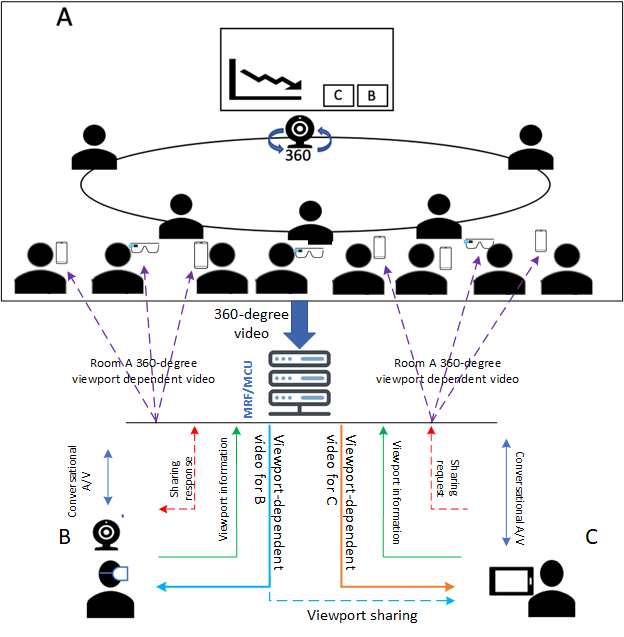


Figure 4.3.2 Proposed viewport sharing use case via MRF/MCU

Figure 4.3.2 shows a second scenario, the call is setup using a MEF or MCU. In this case, the viewport dependent streaming may be handled by MRF or MCU. For the participants with their own display device, the display device may receive 360-degree viewport-independent video from omnidirectional camera, or viewport-dependent video by sending orientation information to MRF/MCU.

## 4.4 Use Case Extension 3: Viewport sharing for second display device

When a remote participant B or C is presenting or communicate to the room A, participants in room A with their own display device may be interested in B or C’s focus or interaction on specific viewport of 360-degree video captured by omnidirectional camera, these participants may request to follow B or C’s viewport and receive the same viewport-dependent stream sending to B or C.

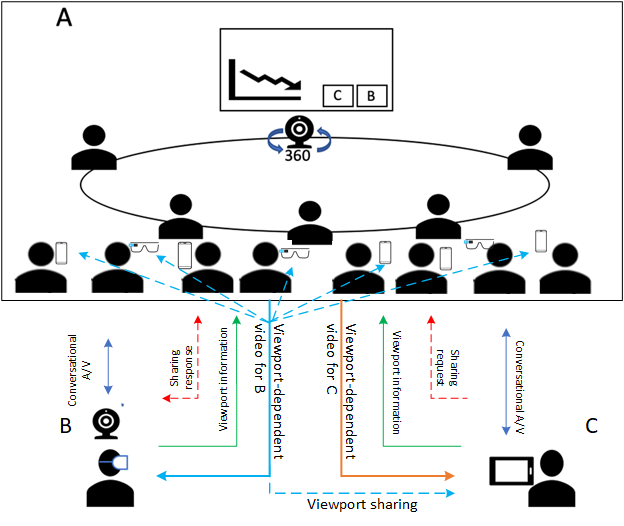


Figure 4.4.1 Viewport sharing to second display w/o MRF/MCU

Figure 4.4.1 shows the first scenario, the call is setup without MEF or MCU. Participants in room A with their own display device may like to follow B’s viewport, participant sends the request to B and is authorized by B. The viewport-dependent video for B is then played back on authorized participant display device with embedded viewport information, the display device would follow embedded viewport information to render the viewport regardless participant’s orientation.

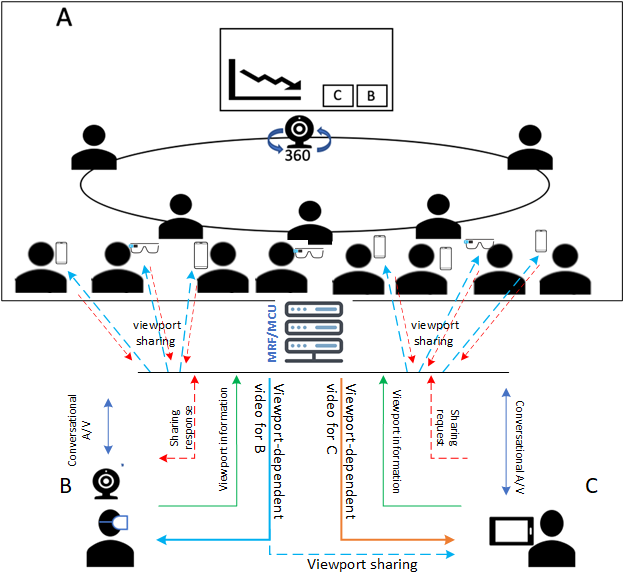


Figure 4.4.2 Proposed viewport sharing use case via MRF/MCU

Figure 4.4.2 shows a second scenario, the call is setup using a MEF or MCU. In this case, the viewport sharing request may be handled by MRF or MCU, MRF or MCU receive the request from participants and confirm the authorization from B. MRF or MCU then sends viewport-dependent video streams with embedded B’s viewport metadata to B and her followers in room A.

## 4.5 Use Case Extension 4: Separate stream for presentation content

John (a participant in conference room A) shares his current computer screen to present his work. This presentation audio/video stream is displayed in the conference room A, but also broadcasted (i.e., sending the same information using unicast to different receivers as in usual MTSI/Telepresence, same for the two instances of "broadcasted" below in this subclause and in subclause 3.5) to all other remote participants (in other conference rooms and for the VR users). While john is still presenting Alice (who is also in the conference room A) adds her screen to be shared with supportive materials for John’s presentation, resulting into two streams being displayed in the room and broadcasted to all other remote participants.

## 4.6 Use Case Extension 5: Applications to 3GPP FLUS

MTSI-based FLUS as defined in TS 26.238 [10] relies on the use of RTP/RTCP-based protocols as defined in TS 26.114. Immersive video and immersive audio/speech may be delivered uplink using MTSI-based FLUS. Thus, even though FLUS is outside the scope of the current ITT4RT work item, the normative specification resulting from the ITT4RT work item could be (and are not required to be) reused in providing the immersive video and immersive audio/speech support for MTSI-based FLUS including (i) recommendations of audio and video codec configurations to deliver high quality VR experiences, (ii) constraints on media elementary streams and RTP encapsulation formats, (iii) recommendations of SDP configurations for negotiating of immersive video and voice/audio capabilities and (iv) RTP/RTCP-based signalling for indication of viewport information to enable viewport-dependent media processing and delivery. In that regard, potential solutions described in clauses 6 and 9 could also be applicable for live uplink streaming scenarios using MTSI-based FLUS.

For immersive video support over MTSI-based FLUS, both in-camera stitching and network-based stitching can be considered. In case of camera stitching, stitched immersive video is sent from the FLUS source (i.e., MTSI sender) to the FLUS sink (e.g., MTSI receiver, which is either in the UE, e.g., as a remote viewer, or in the network, e.g., as a media gateway). This is where immersive video gets delivered uplink using MTSI-based FLUS, and the normative specification output of ITT4RT becomes relevant. In case of network-based stitching, different 2D captures are sent from the FLUS sources (MTSI senders) to the FLUS sink (MTSI receiver as a media gateway) in the network and the media gateway server performs decoding, stitching, and re-encoding to produce the immersive video, which is then distributed to the remote viewers using MTSI or other means. In this case, only 2D video gets delivered using MTSI-based FLUS and immersive video is only delivered in the non-FLUS portion of the link, i.e. downlink. If MTSI-based means are used to deliver the immersive video to the remote viewers, only then the normative specification output of ITT4RT could be applicable.

## 4.7 Use Case Extension 6: Multiple Overlays

### 4.7.1 Scenario 1



Figure 4.7.1.1 Multiple Overlays - Scenario 1

A user is streaming an immersive video, from a teleconference room A. Room A uses screen share to display two streams (overlay 1, overlay 2).

* The sender needs to send information about the overlays such as number of overlays, layering, priority etc to the user via SDP offer.
* The user based on its resource availability may decide if he/she wants to receive all the overlays or just a subset of overlays. This is conveyed back to the sender via SDP answer.
* The sender needs to inform the user if he/she is allowed to use multiple overlays. It may be possible that sender might not want the receiver to stream multiple overlays to avoid possible distractions. One such scenario is when multiple overlay videos are available and the user may be allowed to stream just one of the overlays.

### 4.7.2 Scenario 2



Figure 4.7.2.1 Multiple Overlays - Scenario 1

This scenario consists of multi-party teleconference. A user is streaming an immersive video, from a teleconference room A. Room A uses screen share to display one stream (overlay 1). The overlay from room B is not shared by A. For the user to stream overlays which are not shared by A:

* Firstly, the sender needs to inform the user if he/she is allowed to use multiple overlays as described in scenario 1.
* Secondly, the sender of the 360-degree video should inform the user at the receiver if the user is allowed to use overlays from streams other than the ones shared by the sender. This is based on the content type of the overlays shared by A. The rationale behind the sender asserting control is that the sender might not want the receiver to mix overlays which may possibly cause a distraction to the user.
* Thirdly, the sender should inform the user if he/she is allowed to overlap overlays. A possible example may be when the sender side is presenting and does not want other overlays to overlap with its presentation stream.
* Lastly if the user’s resource availability is limited, default preference maybe given to the sender’s overlay as compared to streaming overlays from other sources.

## 4.8 Use Case Extension 7: Audio mixing of multiple streams in ITT4RT

### 4.8.1 Use Case

The assumptions are as the following:

1. The sender’s conference room has a 360 video with one audio stream (In reality, it may have multiple streams but for simplicity of the usecase, we only consider one audio stream).
2. The sender may have one or more overlays each of which may have its own audio stream.
3. The receiver receives the audio stream from the sender’s room as well as the audio streams from one or more overlays.

The above assumption is shown in the following figures:

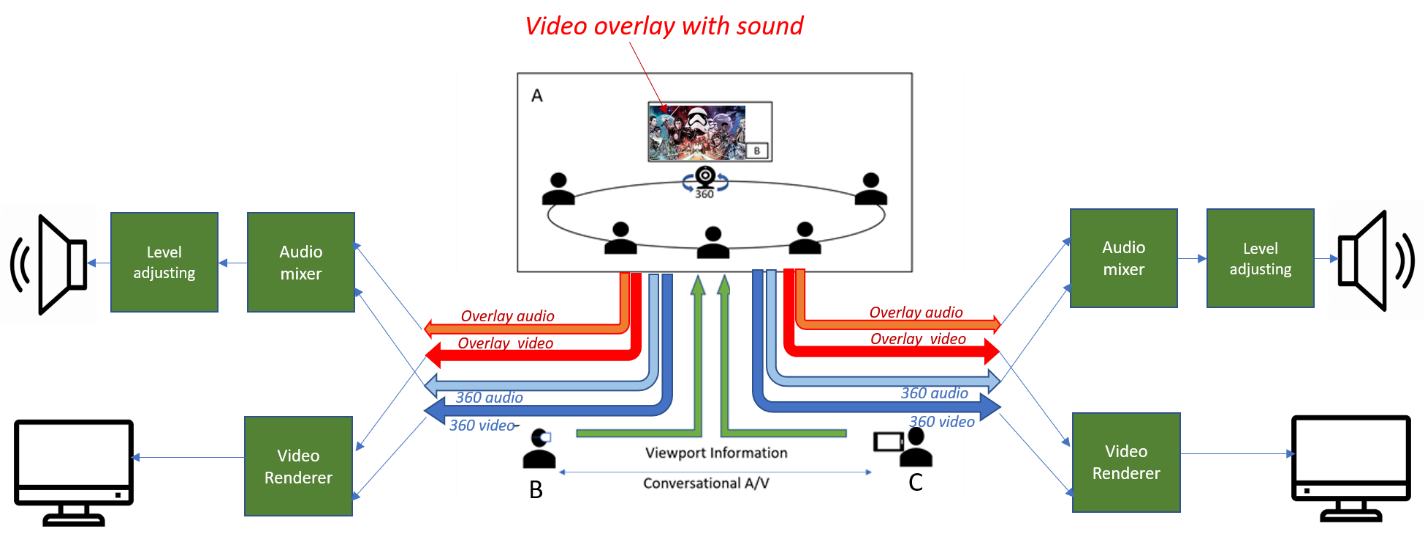


Figure 4.8.1.1 - 360-degree conference call

Note that as is shown in the figure the overlay video has its own overlay audio. Its audio either has to be mixed at the sender with the room’s audio and sent as a single stream, or it needs to be delivered as a separate stream. If it is mixed in the sender, the sender needs to decode the overlay’s audio, mix it with the room’s audio and encode the mix again for transmission. If it is sent as a separate streams, then it should be mixed with the room’s audio at the receiver. This contribution focuses on the audio mixing at the receiver.

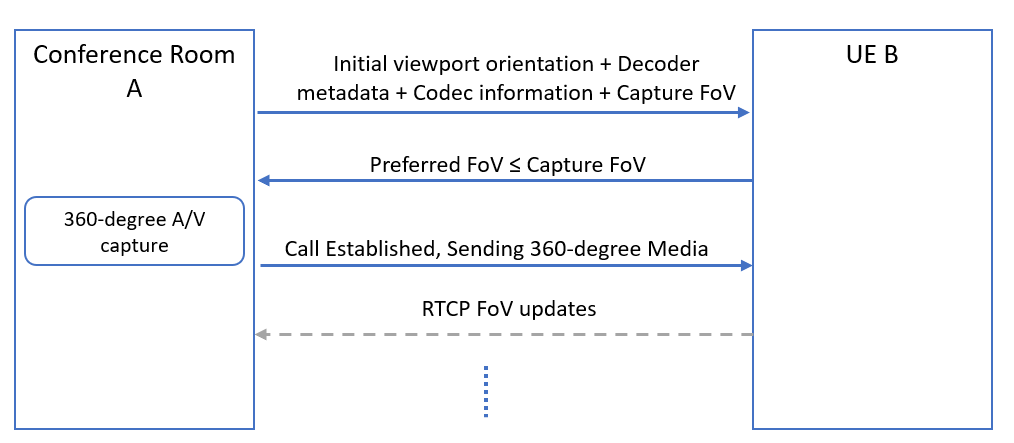
The following use-cases are considered for audio mixing:

* Case 1: Providing the recommended mixing gain at the receiver’s end for rooms and overlay audios.
* Case 2: Changing the recommended mixing gain by sender during a portion of the session to dominate the overlay audio (e.g. reducing the room’s chatter when the overlay audio is important) or the room audio (e.g. when speaker in the room wants the remote audience focus on his speech)

# 5 Requirements

* Multiple single-user participants are supported.
* Communications between the single users can be conventional MTSI/Telepresence communications. MSMTSI could be used, and if that is used, then media data can be transmitted in separate media streams, and the layout of different participants is up to the client application/implementation.
* One 360 camera per location in multi-party conference scenarios involving multiple physical locations are allowed.
* Both in-camera stitching and network-based stitching are supported.
  + In case of camera stitching, stitched immersive video is sent from the conference room to the conferencing server (e.g., MSMTSI MRF or any other media gateway) and then from the conferencing server to the remote participants. If this is a one-to-one conversational session between the conferencing room and the remote participant, a media gateway in the middle might not be necessary.
  + In case of network-based stitching, different 2D captures are sent from the conference room to the conferencing server and the conferencing server performs decoding, stitching, and re-encoding to produce the immersive video, which is then distributed to the remote participants.
* It is recommended that MTSI and IMS Telepresence endpoints support codec, protocol and transport capabilities relevant for encoding, delivery and consumption of immersive speech/audio and immersive video.
* Capability for the party that sends 360-degree video to send viewport-dependent and/or viewport-independent streams.
* Timely delivery of the changes in viewport orientation from the remote participants, and appropriate low-delay actions to update the viewport-dependent streams. Any changes in viewport orientation should not lead to latency-prone signalling, such as SIP renegotiations.
* Capability to create viewport-dependent streams for individual UEs including an larger area of the original viewport for safe playback in the UE.
* A suitable coordinate system to be used as the standard way of communicating the orientation of the viewport.
* Given possible end device limitations as well as potential constraints on the conference room equipment, network-based processing should be considered for media workloads involving both conference room and remote participants, e.g., stitching of captured streams from the conference room, media composition, transcoding and prerendering for the remote participant, etc.
* The following parameters need to be signalled in the SDP during call setup in addition to normal MTSI call signaling [7].
  1. Initial viewport orientation. It is the default orientation from which to start the view at the receivers’ side.
  2. Decoding/Rendering metadata, e.g., region-wise packing information, projection mapping information, frame packing information, etc. It is subject of discussion whether this information is signaled via SDP and/or within SEI messages with the media stream.
  3. Capture Field-of-View (CFoV): as discussed during the definition of the use case, the system supports transmission of 360-degree video. However, the range of the FoV may be restricted in order to enhance user experience. The negotiation requires signaling the capture FoV of the capture device, and a response carrying the receiver’s preferred FoV (PFoV) depending on the remote UE, where the preferred FoV will be less than or equal to the captured FoV.
  4. Codec negotiation

The high level signaling flows are depicted in Figure 5.1. The user C is not represented here for simplicity, but this is not a restriction for our reasoning. In this example MRF/MCU is not used.



**Figure 5.1 Signalling flow for a 360-degree conference call with unidirectional 360- degree video from A to B.**

* Once the call has been established, remote parties (B or C) can send viewport orientation information using RTCP reports with yaw, pitch and roll data. These may be sent at fixed intervals or event-based, triggered by changes in viewport orientation. or hybrid combination of fixed interval and event-based triggers. When hybrid reporting scheme is used, the event-based feedback is triggered by any changes in viewport whereas the regular RTCP interval provides the sender with a regular update, in fixed intervals, of the viewport even if the event-based feedback is not received. The most efficient RTCP reporting scheme for viewport orientation information is for further study.
* Capability to support the interaction where all media types will be presented to certain users and a subset of the media types are presented to the others.
* Capability for the participant in room A with his or her own display device to receive a viewport independent or viewport dependent video from omnidirectional camera in room A.
* Capability for the remote party to share a viewport dependent video stream with embedded viewport metadata to another remote participant.
* Capability for the participant in room A with his or her own display device to follow remote participant viewport presentation.
* The capability to place overlays in the 360-degree video either within the device or pre-rendered through a network element.
* Transmission from sender to receiver of the coordinates of the location of the overlay (e.g. a presentation): this is a necessary and basic requirement that will give flexibility in the overlay placement at the receiver’s side. By sender/receiver it is meant either one of the parties or the MRF/MCU.
* Avoid that the overlaid background content is transmitted unnecessarily at high quality within the user viewport: this is a basic issue that overlays cause to viewport-dependent streaming. The content in the viewport is always streamed at higher quality. However, when an overlay with different content is sticked on top of (part of) the viewport, the content behind the overlay does not need to be sent ay higher quality. This allows saving bandwidth or increase the quality in the viewport for the non overlaid parts.
* Enable some form of interaction with the overlay (e.g., moving or rotating the overlay, resizing it, switching it on/off, etc.): these are basic and simple ways to interact with the overlay, to increase flexibility and utility of an overlay.
* Capability for users to receive an incoming interaction message (e.g., SMS, chat message, voice call or audio-visual call) from other users as an overlay: this is a good way to allow integration of other 3GPP services and applications into ITT4RT/MTSI applications in order to increase the value of the first VR applications for 3GPP.
* To facilitate network-based stitching, it is possible to signal camera calibration parameters for each 2D video capture (i.e., each camera lens) transported from the conference room to the conferencing server at the beginning of each session. Relevant intrinsic and extrinsic camera parameters can include lens numbers, layouts, positions, angles, radius, distortion, entrance pupil and resolutions.
* An RTP receiver should be able to signal higher-level metrics such as Motion to High-Quality Delay to the sender to assist in bandwidth adaptation and monitoring.
* Allow still background images to be used when network conditions do not permit transmitting a video stream for the area outside the viewport.
* A sender can offer support for conditional overlays in session signalling and receiver can understand it.
* To activate/deactivate conditional overlays, a receiver is capable to signal one or more regions and their associated conditions using RTCP feedback/SDP to the sender during a media session.
* Capability to identify the location of the presentation and where to insert an overlay of the alternative presentation into the omnidirectional content:
* while stitching the different camera images together in the network
* after the stitching of the camera images into the omnidirectional video (e.g. by reencoding the omnidirectional video in the network)
* after receiving the stitched omnidirectional video and the overlay by the receiving client
* If the overlay is represented in video format, it **should** be delivered over RTP. If the overlay is represented in formats other than video (e.g., images), then the overlay may be delivered considering the following formats and protocols:
* the overlay may be delivered using the WebRTC data channel (SCTP/DTLS) [11], which is currently already defined for both MTSI in TS 26.114 and IMS-based Telepresence in TS 26.223.
* the overlay may be represented according to the HEVC image format as specified in Annex B of [12]. Accordingly HEIF image items and image sequences are described as regular HEVC media streams. It should be possible to use the HEVC Payload format as described in RFC 7798 for the delivery of the HEVC images, image collections, and image sequences (details TBD). All NAL units of an HEVC image stored as meta items are extracted and transmitted as HEVC access units with the same presentation time stamp.
* Capability to signal one or more regions of the 360 degree video over which overlay media should not be completely or partially rendered at the receiver side. The provided information can be used when a receiver is rendering viewport-relative overlays over its selected viewport.

# 6 Architecture

## 6.1 General

The current MTSI service architecture depicted in Figure 4.1 of TS 26.114 [7] is applicable for immersive teleconferencing. No further architectural gaps are identified.

In terms of the reuse of existing MTSI functionality, the following may be observed:

1. For in-camera stitching, stitched immersive video is sent from the conferencing room to the conferencing server (e.g., MSMTSI MRF) or directly to the remote participant (e.g., one-to-one conversation) in one or more RTP streams (e.g., established via SDP). Multiple RTP streams may be used in case tile or sub-picture based delivery optimization is in use.
2. For network-based stitching, multiple RTP streams are established (e.g., via SDP, using MSMTSI) between the conferencing server and conference room, each of which carries a particular 2D capture. These RTP streams are then sent from the conference room to the conferencing server and the conferencing server performs decoding, stitching, and re-encoding to produce one or more RTP streams containing the immersive video, which are then distributed to the remote participants (e.g., again via MSMTSI). Multiple RTP streams may be used for the immersive video in case tile or sub-picture based delivery optimization is in use.

## 6.2 Potential architecture extensions

### 6.2.1 Network-based Media Processing

#### 6.2.1.1 Introduction

MPEG Network-based Media Processing (NBMP) in ISO/IEC 23090-8 [13] may be used to establish media processing tasks for network-based stitching at MRF/MCU sink to be performed on received media components from an ITT4RT-Tx client. In particular, the NBMP Workflow Description Document for the 360-degree video stitching workflow may be provided to the MRF/MCU, which would then use the information during negotiation of the media streams to be used in the session and subsequent media plane processing. In addition to 360-degree video stitching, NBMP can also be used for initiating other media processing workflows in MRF/MCU such as guided transcoding, overlaying on image or video backgrounds or viewport-dependent content customization.

#### 6.2.1.2 Example Architecture

An example architecture for establishing an NBMP workflow for ITT4RT is shown in Figure 6.2.1.1. In this case the NBMP client resides in the AS, and is responsible for providing the NBMP workflow description to the MRF, which contains the NBMP workflow manager. The MRF would establish the necessary media flows for accessing the media from the ITT4RT-Tx client (NBMP media source). The required tasks (e.g. stitching, VDP etc. ) can be set up in the MRF before delivering the processed media to the ITT4RT-Rx client (NBMP media sink). The establishment of media flows, e.g., unstitched video from an ITT4RT-Tx client, viewport-dependent 360-degree video to an ITT4RT-Rx client, is done in accordance with MTSI specification by the MRF. The AS is aware of the capabilities and address of the MRF. Further details on workflow description document exchange is considered out-of-scope of this document. If further normative work is needed for Mr,Cr interface between the AS and MRF to support NBMP is left FFS.

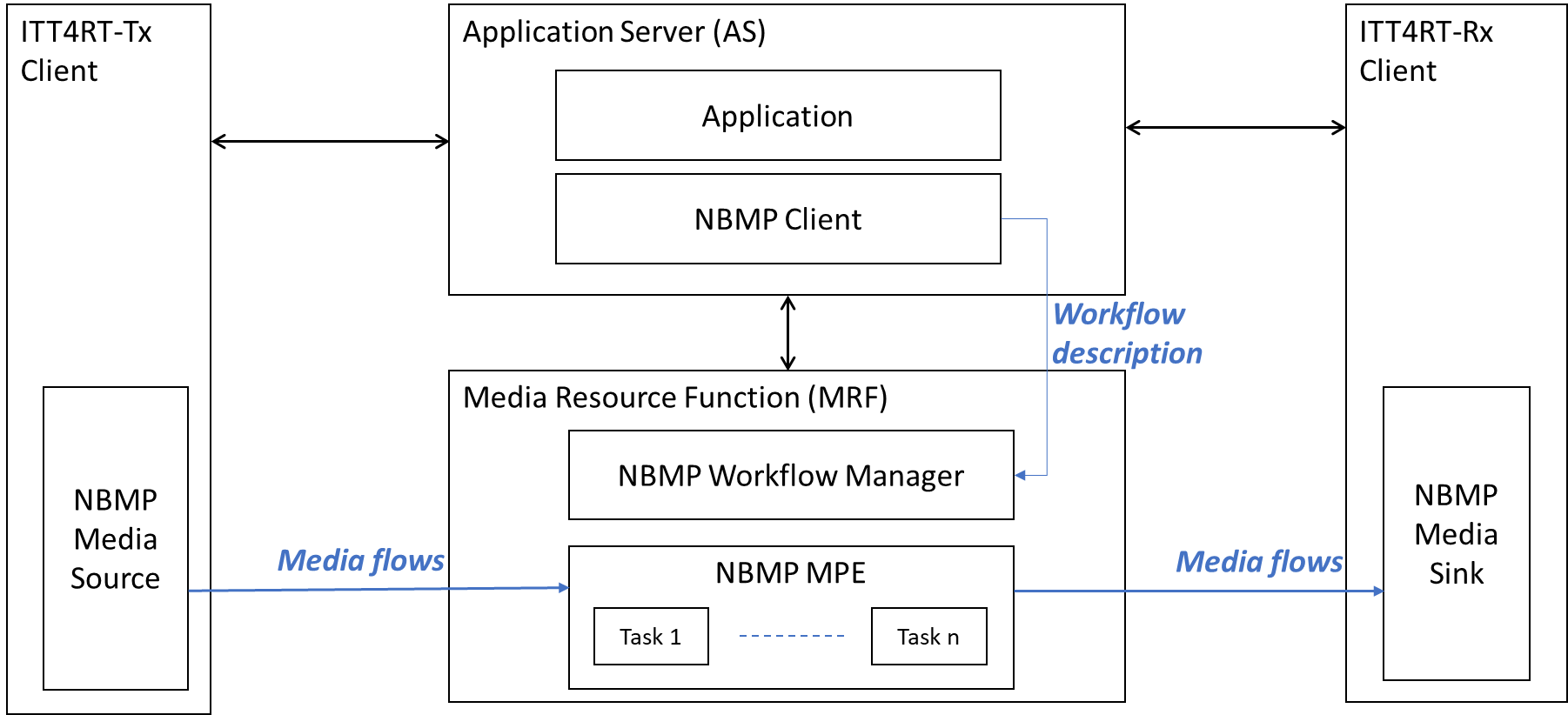


Figure 6.2.1.1 An example architecture for MPEG NBMP for ITT4RT media processing

# 7 Potential Solutions

## 7.1 Frame packing of overlays

Frame packing can be used when delivering overlays as a single RTP stream. The overlay source with frame packed overlays can be:

1. an RTP stream with an overlay source frame packed with a 360-degree video or with another overlay such that it is decodable by a single decoder.
2. an RTP stream with one or more overlay sources; frame packing is used for more than one overlay source.

Overlays frame-packed with 360-degree video or with other overlay streams require a single decoder, whereas overlay video that is encoded and delivered separately from the 360-degree video and other overlay streams requires multiple decoders and may require synchronization, depending on the use case. However, having a separate overlay stream allows more freedom to use a higher resolution for the overlay. Furthermore, packing the overlay as part of the 360-degree video would also affect the resolution of the 360-degree content.

### 7.1.1 SDP Signalling and Examples

Frame packed stream may be described with the following attribute included in the packed stream’s media description:

a=itt4rt\_framepacking: <id1> <id2> <id3> …. <PPC>

The list of ids is an identifier for the source media packed in the stream. If the attribute is omitted in the response, the stream consists of only the source identified by the first identifier. The ids are the mid values in the media description of the individually encoded media streams of the same media source. The parameter PPC is a binary value that defines the packed picture content:

Table 7.1.1.1: PPC parameter

|  |  |
| --- | --- |
| PPC (Binary Value) | Packed Streams |
| 001b | Only overlay |
| 010b | Only 360-degree content |
| 011b | 360-degree content + overlay |

The attribute itt4rt\_framepacking may be used for identifying packed overlays even when individually encoded streams are not offered for those overlays. In this case the ids will not be the mid values, but the overlay\_id described in the 3gpp\_overlay attribute. The ids would appear in the order they appear in the RHE so that each can be associated with the packed stream.

In the following example in Figure 7.1.1.1, an overlay is offered as an individual stream with the mid B. The same overlay is also offered as frame packed with the 360-degree video with mid A as indicated by the itt4rt\_framepacking attribute.

|  |
| --- |
| SDP offer |
| a=itt4rt\_group: A B  m=video 49156 RTP/AVP 100  a=tcap:1 RTP/AVPF  a=pcfg:1 t=1  a=rtpmap:100 H265/90000  a=3gpp\_360video:100… **/\*360-degree video configuration\*/**  a=itt4rt\_framepacking: A B PPC=0x03 **/\*media with mid B and mid A are framepacked in this media stream and the packed stream consists of 360-degree content and overlay\*/**  a=3gpp\_overlay:B …**/\*configuration of the overlay \*/**  a=imageattr:100 send [x=7680,y=4320]  a=mid:A  m=video 49158 RTP/AVP 99  a=tcap:1 RTP/AVPF  a=pcfg:1 t=1  **/\*overlay video of closeup of presentation in room A (spherical)\*/**  a=rtpmap:99 H265/90000  a=imageattr:99 send [x=1280,y=720] [x=640,y=480]  a=mid:B |

Figure 7.1.1.1: SDP offer with frame packed overlays

The following SDP answer in Figure 7.1.1.2 excludes the individual encoding of the overlay and includes only the frame-packed stream.

|  |
| --- |
| SDP answer |
| a=itt4rt\_group: A B  m=video 49156 RTP/AVP 100  a=tcap:1 RTP/AVPF  a=pcfg:1 t=1  a=rtpmap:100 H265/90000  a=3gpp\_360video: 100 … **/\*360-degree video configuration\*/**  a=itt4rt\_framepacking: A B PPC=0x03 **/\*media with mid B and mid A are framepacked in this media stream and the packed stream consists of 360-degree content and overlay\*/**  a=3gpp\_overlay:B… **/\*configuration of the overlay \*/**  a=imageattr:100 send [x=7680,y=4320]  a=mid:A  m=video 0 RTP/AVP 99  a=tcap:1 RTP/AVPF  a=pcfg:1 t=1  **/\*overlay video of closeup of presentation in room A (spherical)\*/**  a=rtpmap:99 H265/90000  a=imageattr:99 send [x=1280,y=720] [x=640,y=480]  a=mid:B |

Figure 7.1.1.2: SDP answer accept for frame packed overlays

An alternative answer is shown below in Figure 7.1.1.3, in which the overlay stream is negotiated to be delivered individually encoded. In this case the itt4rt\_framepacking attribute is removed from the 360-degree video media description so the overlay is not frame packed.

|  |
| --- |
| SDP answer |
| a=itt4rt\_group: A B  m=video 49154 RTP/AVP 100  a=tcap:1 RTP/AVPF  a=pcfg:1 t=1  a=rtpmap:100 H265/90000  a=3gpp\_360video:100… **/\*360-degree video configuration\*/**  a=3gpp\_overlay:B … **/\*configuration of the overlay \*/**  a=imageattr:100 send [x=7680,y=4320]  a=mid:A  m=video 49158 RTP/AVP 99  a=tcap:1 RTP/AVPF  a=pcfg:1 t=1  **/\*overlay video\*/**  a=rtpmap:99 H265/90000  a=imageattr:99 send [x=1280,y=720]  a=mid:B |

Figure 7.1.1.3: SDP answer reject for frame packed overlays

### 7.1.2 Bitstream Signalling

NOTE: Whether RTP header extensions provide a feasible method for bitstream signalling or if a different mechanism can be used for this purpose is FFS.

The RWP SEI message may be used to carry the packing information. However, it does not define the role of the packed regions i.e., 360-degree video and overlay, nor does it indicate the layering order. Therefore, an RTP header extension (RHE) can be considered for carrying the packing information (packed and projected regions), layering order and type of stream (overlay/360-degree video).

The packed frame can be delivered with RTP header extension as shown in Figure 7.1.2.1 (this example shows two packed regions: one overlay region and one 360-degree video):

0 1 2 3  
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
| ID | len=54 |F|N\_Regions=n| QR |LYR| TT |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
| Projected\_Region\_Width |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
| Projected\_Region\_Height |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
| Projected\_Region\_Top |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
| Projected\_Region\_Left |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
| Packed\_Region\_Width | Packed\_Region\_Height |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
| Packed\_Region\_Top | Packed\_Region\_Left |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
|F| QR |LYR| TT | Projected\_Region\_Width |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
|Projected\_Reg\_Widt | Projected\_Region\_Height |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
|Projected\_Reg\_Heig | Projected\_Region\_Top |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
|Projected\_Reg\_Top | Projected\_Region\_Left |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
|Projected\_Reg\_Left | Packed\_Region\_Width |Pkd\_Rg\_Ht |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
|Packed\_Reg\_Ht | Packed\_Region\_Top |Pkd\_Rg\_Left|  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
|Packed\_Reg\_Left | Zero pad |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure 7.1.2.1: Potential RTP Header extension for frame-packed overlays

Projected\_Region\_Width (or Pkd\_RegWid), Projected\_Region\_Height (or Pkd\_Rg\_Ht), Projected\_Region\_Top and Projected\_Region\_Left are the width, height, top offset and left offset, respectively, of the projected region. Packed\_Region\_Width, Packed\_Region\_Height, Packed\_Region\_Top and Packed\_Region\_Left are the width, height, top offset and left offset of the packed region.

The 8-bit ID is the local identifier as defined in [7]. The length field defines that 54 bytes follow (in this example). The field N\_Regions gives the total number of packed regions (n), which is 2. The value QR is the index of the packed region that follows in this extension header. The index of the picture (QR) is assigned based on quality, with the highest quality picture always being the first (0) and lowest one being the last (n-1). The LYR field provides the layer of the packed region. This is the same as the layering order defined for OMAF. In case of one 360-degree video and one overlay, the N regions will be two, with LYR = 0 for background and LYR = 1 for overlay. The TT field provides the transform type which is one of the following:

0: no transform

1: mirroring 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)

In case of more than one layer, the layers are included in ascending order with the lowest layer described first. If more than one region belongs to a single layer, the multiple regions for the same layer occur consecutively in ascending order of the QR value.

## 7.2 Conditional overlays

### 7.2.1 Introduction

According to the MPEG OMAF specification SC29WG11\_N19435, a sender may have a recommended viewport (as defined in section 7.14.3.1) as an area of interest to be displayed as an overlay. The overlay can be treated as conditional by the receiver and displayed only when the area is not within the viewport, as defined in Section G.5, in this case using a sender specified “condition”.

In ITT4RT, the same concept can be used to receive an overlay only when a certain condition is met. The region can be defined by the receiver UE instead of the sender’s recommended viewport as in the case of OMAF to cater to the conversational and real-time aspects.

A conditional overlay is defined as an overlay that is activated (i.e., transmitted by the sender) as an overlay media stream when a certain condition is met, and deactivated when the condition is no longer met. RTP stream of an overlay can be activated or paused by an ITT4RT-Rx client using pause/resume as specified in TS 26.114 for this purpose.

The following solution may be of relevance when overlays are frame packed with other media or if the ITT4RT-Rx client wants the control of the overlay activation/deactivation to be at the ITT4RT-Tx client.

In this case, the capability is advertised by the ITT4RT-Tx client to the ITT4RT-Rx clients using SIP/SDP. Possible use cases where a conditional overlay may be employed (the overlay activation condition is described in italic):

* displaying a region of interest of the 360-degree video when the viewport is not overlapping with that region of interest. The capability may be used for extending the FoV for *devices when two areas of interest do not fit within the FoV of the device.*

### 7.2.2 Potential solution using RTCP

A solution for the use of conditional overlays is described below.

* The ITT4RT-Tx client indicates the capability to support conditional overlays using the SDP field a=rtcp-fb:\* 3gpp-conditional-overlay. The field indicates that the ITT4RT-Tx client is capable of receiving RTCP feedback messages from the ITT4RT-Rx client indicating the region for enabling a conditional overlay.
* The ITT4RT-Rx client indicates in an RTCP conditional overlay feedback packet the region of the sphere it wishes to receive as a conditional overlay, along with the condition associated with it. The RTCP feedback has the following fields:
  + Conditional\_overlay id
  + Azimuth of the center of the sphere region with respect to the origin of the omnidirectional content coordinate system. The unit is expressed in degrees in the range of [0, 360].
  + Elevation of the centre of the sphere region with respect to the origin of the omnidirectional content coordinate system. The unit is expressed in degrees in the range of [0, 180].
  + Azimuth range specifies the azimuth range of the conditional overlay through the center point of the sphere region. The unit is expressed in degrees in the range of [0, 360].
  + Elevation range specifies the elevation range of the conditional overlay through the center point of the sphere region. The unit is expressed in degrees in the range of [0, 180].
  + The condition for activating the conditional overlay defined as the maximum ratio of overlap between the conditional region defined above and the viewport for it to be activated. For example, if the value is 0; this implies that if the ratio of the portion of the conditional overlay region covered by the viewport to the size of the conditional overlay is 0, then the overlay is activated (no overlap).
* The ITT4RT-Tx client will deliver the indicated region as an overlay stream to the ITT4RT-Rx client when the condition is met. An RTP header extension can be used to indicate the actual region that is transmitted as overlay to the ITT4RT-Rx client depending on sender limitations. The support for this extension is indicated in the SDP using a=extmap:7 urn:3gpp:cond-overlay-sent.
* The ITT4RT-Rx client may cancel conditional overlays by sending a cancellation RTCP conditional overlay message with a list of the appropriate conditional overlay ids.

An example condition may be:

* Condition: 0.5. Upon receiving this RTCP message, the ITT4RT-Tx client will respond by activating the overlay if the overlap of the conditional region and the viewport is between 0 and 50% at any time until the conditional overlay is either cancelled or the session ends.

### 7.2.3 Potential solution using SDP

An ITT4RT-Tx client may, in its SDP offer, indicate the capability to support conditional overlays for a particular 360 video using the attribute a=3gpp\_conditional\_overlay. The ITT4RT-Rx client includes the same attribute in the SDP answer to show it also supports conditional overlay.

Once the media stream is established, the ITT4RT-Rx client may select a region of interest it wants to receive as an overlay. It can then signal to the ITT4RT-Tx client using the SDP offer of the same 360-degree video but indicating within the request:

a = 3gpp\_conditional\_overlay:<azimuth, azimuth\_range, elevation, elevation\_range, condition>

The parameters azimuth, azimuth\_range, elevation, elevation\_range and condition are the same as previously defined for the RTCP solution. An example for a conditional overlay request is : a = 3gpp\_conditional\_overlay: 0, 90, 0, 45, 0.5; requesting an overlay of the region centered at the global center 0,0 with the dimensions 90 degrees horizontally and 45 degrees vertically, which is transmitted whenever the overlap is between 0 and 50%.

# 8 Selected solutions in TS 26.114

This section lists the selected solutions as presented in TS 26.114.

Y.2 **ITT4RT Reference architecture**: A reference architecture is presented that includes a sender and receiver architecture to showcase the basic interfaces and functional blocks required to build ITT4RT clients.

Y.3 **Immersive 360-Degree Video Support**: Lists the different decoding capabilities an ITT4RT client needs to support. This also includes details on mapping information related to 360-degree video data.

Y.4 **Immersive Voice/Audio Support**: Currently, an ITT4RT client needs to support any existing audio solution (i.e., clause 5.2.1 of TS 26.114). We expect further updates on ITT4RT audio capabilities (including immersive audio) in the future.

Y.5 **Overlay Support**: ITT4RT supports both 2D and spherical overlays. Further details on overlay and media configuration are presented in Y.6.

Y.6 **Media configuration**:Multiple media configurations have been selected, including new SDP attributes and scene description-based overlay solutions, to specify functions and negotiations between ITT4RT clients.

Y.7 **Media transport**: The selected solution for ITT4RT media transport includes RTP and RTCP transmission.

Y.8 **SDP Examples**: SDP examples have been selected as informative additions to the selected solutions.

Y.9 **Recommended audio mixing gains**: The ITT4RT-Tx client can specify recommended gains for mixing of its transmitted audio streams.

Annex A (informative):  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2022-02 | SA4#117-e | S4-220258 |  |  |  | Drafting and integration of the ITT4RT PD | 0.1.0 |
| 2022-02 | SA4#117-e | S4-220265 |  |  |  | Cleanup of document including fixing of references | 0.2.0 |

1. The system supports transmission of full 360 video. However, the use cases may restrict the field of view to enhance user experience. [↑](#footnote-ref-1)