**3GPP TSG- Meeting # *XXX***

**, , - revision of S4-241123**

|  |
| --- |
| *CR-Form-v12.3* |
| **CHANGE REQUEST** |
|  |
|  |  | **CR** |  | **rev** | 3 | **Current version:** |  |  |
|  |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME | **x** | Radio Access Network |  | Core Network | **x** |

|  |
| --- |
|  |
| ***Title:***  |  |
|  |  |
| ***Source to WG:*** |  |
| ***Source to TSG:*** |  |
|  |  |
| ***Work item code:*** |  |  | ***Date:*** |  |
|  |  |  |  |  |
| ***Category:*** |  |  | ***Release:*** |  |
|  | *Use one of the following categories:****F*** *(correction)****A*** *(mirror corresponding to a change in an earlier release)****B*** *(addition of feature),* ***C*** *(functional modification of feature)****D*** *(editorial modification)*Detailed explanations of the above categories canbe found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | *Use one of the following releases:Rel-8 (Release 8)Rel-9 (Release 9)Rel-10 (Release 10)Rel-11 (Release 11)…Rel-17 (Release 17)Rel-18 (Release 18)Rel-19 (Release 19) Rel-20 (Release 20)* |
|  |  |
| ***Reason for change:*** | Ambiguities in the RTP Payload Format have been observed. While addressing those, several corrections to the syntax and SDP parameters had to be introduced. |
|  |  |
| ***Summary of change:*** | CMRs for IVAS were added to allow rate/format/bandwidth adaptationE byte was defined for IVAS operation Added PI data definition in IVAS payload and PI data type detailsClarification on ivas-mode-switch, coded format and other SDP parametersAddition of SDP offer-answer proceduresClarifications on terminologyClarifications on implementation complexityVarious other minor fixes |
|  |  |
| ***Consequences if not approved:*** | No successful SDP negotiation and RTP connection between two endpoints. |
|  |  |
| ***Clauses affected:*** | 2, Annex A |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** | **X** |  |  Other core specifications  | CR26114-0561 |
| ***affected:*** |  | **x** |  Test specifications |  |
| ***(show related CRs)*** |  | **x** |  O&M Specifications |  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** | Initial version: S4-240664 (SA4#127-bis-e endorsed)Rev 1: Implementation of several clarifications including PI data and SDP offer-answer proceduresRev 2: Clarification on ivas-mode-switch and evs-mode-switch handlingRev 3: Restructuring of new text and clean-up |

CHANGE 1

# 2 References

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

[2] 3GPP TS 26.441: "Codec for Enhanced Voice Services (EVS); General Overview".

[3] 3GPP TS 26.445: "Codec for Enhanced Voice Services (EVS); Detailed Algorithmic Description".

[4] 3GPP TS 26.447: "Codec for Enhanced Voice Services (EVS); Error concealment of lost packets".

[5] 3GPP TS 26.448: "Codec for Enhanced Voice Services (EVS); Jitter Buffer Management"

[6] 3GPP TS 26.250: "Codec for Immersive Voice and Audio Services (IVAS); General overview".

[7] 3GPP TS 26.251: "Codec for Immersive Voice and Audio Services (IVAS); C code (fixed-point)".

[8] 3GPP TS 26.252: "Codec for Immersive Voice and Audio Services (IVAS); Test Sequences".

[9] 3GPP TS 26.254: "Codec for Immersive Voice and Audio Services (IVAS); Rendering".

[10] 3GPP TS 26.255: "Codec for Immersive Voice and Audio Services (IVAS); Error concealment of lost packets".

[11] 3GPP TS 26.256: "Codec for Immersive Voice and Audio Services (IVAS); Jitter Buffer Management".

[12] 3GPP TS 26.258: "Codec for Immersive Voice and Audio Services (IVAS); C code (floating point)".

[13] C. de Boor and K. Höllig (1987), B-splines without divided differences, in Geometric Modeling, G. Farin ed., SIAM, 21–27.

[14] Borß, C. A Polygon-Based Panning Method for 3D Loudspeaker Setups. In *Audio Engineering Society Convention 137*, Los Angeles, USA, Oct. 2014.

[15] Zotter, F. and Frank, M., All-Round Ambisonic Panning and Decoding, *J. Audio Eng. Soc.*, vol. 60, no. 10, pp. 807-820 (Oct. 2012)

[16] Allen, J. B., & Berkley, D. A., Image method for efficiently simulating small room acoustics. *Journal of the Acoustical Society of America*, 65(4), 943-950, (1979).

[17] Box, G. E. P. and Cox, D. R. (1964). An analysis of transformations, Journal of the Royal Statistical Society, Series B, 26, 211-252.

[18] 3GPP TS 26.118: “Virtual Reality (VR) profiles for streaming applications”.

[19] Ivanic, J., & Ruedenberg, K., Rotation matrices for real spherical harmonics. Direct determination by recursion. *The Journal of Physical Chemistry*, *100*(15), 6342-6347, 1996.

[20] C. R. Helmrich and B. Edler, “Signal-Adaptive Transform Kernel Switching for Stereo Audio Coding,” in Proc. IEEE WASPAA, New Paltz, NY, USA, Oct. 2015.

[21] AES, “AES69-2022: AES standard for file exchange - Spatial acoustic data”, Audio Engineering Society, 2022.

[22] F. Thomas, "Approaching Dual Quaternions From Matrix Algebra," in *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1037-1048, Oct. 2014.

[23] J, Daniel “R,eprésentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimedia”, Thèse de doctorat de l’Université Paris 6, 2001

[24] M. Chapman, “A Standard for Interchange of Ambisonic Signal Sets. Including a file standard with metadata”, Ambisonics Symposium 2009, Graz, June 25-27

[25] ISO/IEC 23091-3:2018 - Information technology Coding-independent code points Part 3: Audio

[26] ISO/IEC 23008-3:2015 - Information technology High efficiency coding and media delivery in heterogeneous environments Part 3: 3D audio

[r1] IETF RFC 4566 (2006): "SDP: Session Description Protocol", M. Handley, V. Jacobson and C. Perkins.

[r2] 3GPP TS 26.114: "IP Multimedia Subsystem (IMS); Multimedia Telephony; Media handling and interaction".

[r3] IETF RFC 3550

[r4] IETF RFC 3551

[r5] IETF RFC 4867

[r6] IETF RFC 7160

[X1] ETSI TS 103 634: " Digital Enhanced Cordless Telecommunications (DECT); Low Complexity Communication Codec plus (LC3plus)"

CHANGE 1

Annex A (normative):
RTP Payload Format and SDP Parameters

# A.1 Introduction

This annex describes a generic RTP payload format and SDP parameters for the Immersive Voice and Audio Services (IVAS) codec for mobile communication [6]. The IVAS RTP packets consist of the RTP header, and the IVAS payload. The IVAS payload consists of IVAS-specific payload header, frame data, and optionally processing information (PI) data.

IVAS is the immersive voice and audio extension of the Enhanced Voice Services (EVS) codec [2], fully incorporating the EVS codec.

# A.2 Conventions, Definitions and Acronyms

## A.2.1 Byte Order

The byte order used in this document is the network byte order, i.e., the most significant byte is transmitted first. The bit order is most significant bit first. This practice is presented in all figures as having the most significant bit located left-most on each line and indicated with the lowest number.

## A.2.2 List of Acronyms

See clause 3.3 for the abbreviations.

# A.3 Payload Format

## A.3.1 Format Overview

The RTP Payload Format described in this document addresses the specific requirements of the IVAS codec. The format supports the transmission of IVAS Immersive mode frames or EVS coded frames with the following features:

- IVAS Immersive mode operation

- WB, SWB and FB audio bandwidths, respectively 16, 32, and 48 kHz sampling rates

- all immersive formats of the IVAS codec

- 1-4 independent (mono) streams with meta data (ISM)

- stereo (including binaural audio)

- multi-channel in 5.1, 7.1, 5.1+2, 5.1+4, 7.1+4

- scene-based audio (Ambisonics) up to order 3 (SBA)

- metadata-assisted spatial audio (MASA)

- combinations of ISM+MASA (OMASA) and ISM+SBA (OSBA)

- bitrates ranging from 13.2 kbps to 512 kbps

- EVS operation

- supporting all EVS operation modes (mono) of the IVAS codec, including the EVS Primary and AMR-WB IO modes, using a payload syntax compatible to the header-full format defined in Annex A of [3] (with some limitations)

NOTE: The format does not support the compact format, present in Annex A of [3].

- NB, WB, SWB and FB audio, respectively 8, 16, 32, and 48 kHz sampling rates

- bitrates ranging from 5.9 (VBR) to 128 kbps

- 20 ms frame duration

- multiple frames per RTP payload, handling of MTU size limits is ffs

- rate adaptation on a per-frame basis; adaptation of bandwidth, format and packetization are ffs

- Discontinuous Transmission (DTX)

- transmission of Processing Information (PI), i.e. PI data, in forward [and reverse] direction to support the rendering[, send requests and feedback]

- switching between EVS (mono) and IVAS (stereo and immersive) operation in the same payload type

## A.3.2 RTP Header Usage

The format of the RTP header is specified in [r3]. This IVAS RTP payload format uses the fields of the RTP header in a manner consistent with the usages in [r3].

The assignment of the RTP payload type for IVAS is out of scope of this document. In most cases SDP would be used to signal the payload type for dynamic assignment.

The RTP clock rate for IVAS is 16000, regardless of the audio bandwidth. A clock rate of 16000 is also used for the AMR-WB [r5] and EVS codecs [3]; having a unique clock rate across all payload types of one media avoids the issues described in [r6].

The RTP timestamp defines the sampling instant (media time) of the first sample of the first IVAS frame in an RTP packet. The duration of one IVAS frame is 20 ms. Thus, the media time is increased for each successive IVAS frame of an RTP packet by 320 ticks. The RTP timestamp of a packet is used for the first PI data in the IVAS RTP payload. The timing of PI frames during DTX is explained in clause A.3.5.x.

The RTP header marker bit (M) shall be set to 1 for the first packet of a talk spurt, i.e. if the first frame-block carried in the RTP packet contains the frame first in a talkspurt. For all other RTP packets the marker bit shall be set to zero (M=0). This is the same usage as described in [r4].

## A.3.3 Packet Payload Structure

### A.3.3.1 General

The IVAS encoder generates encoded frames representing 20 ms of speech or audio data. The IVAS payload contains:

- (optional) E-bytes (including the CMR) for adaptation and indication of optional PI data section;

- one or more ToC(s) describing the IVAS audio frame(s) included in the payload;

- IVAS frame data block(s), representing 20 ms of speech or audio data (depending on ToC signaling), and;

- optional PI data section;

### A.3.3.2 Format Description

An RTP payload comprises the IVAS payload, which consist of the IVAS-specific payload header followed by the frame data and optional PI data as shown in Figure A.3.3.2-1. The frame data consists of one or more IVAS or EVS coded frames (including NO\_DATA, see A.3.3.3.2). The optional PI data section can be considered as additional metadata to support the rendering[, send requests and feedback].There may be zero-padding bits in addition at the end of the payload. Padding bits shall be discarded by the receiver.

NOTE: The purpose of padding is that in the case of EVS AMR-WB IO frames, payload data may need to be octet-aligned using zero-padding bits at the end of the payload. EVS Primary frames are by definition octet-aligned (see clause A.2.2.1.4.1 of [3]).

+-----------------------+---------------------+--------------------+----------+
| RTP Header (+ HDREXT) | payload header | frame data | PI data |
+-----------------------+---------------------+--------------------+----------+

 \--------------------\ /------------------------------/
 IVAS payload

Figure A.3.3.2-1: RTP Header with IVAS payload structure

NOTE: RTP header extensions, RTCP Feedback/APP, Multi-Stream Handling are ffs

### A.3.3.3 Payload Header

#### A.3.3.3.1 General

The IVAS payload header consists of Table of Contents (ToC) bytes and Extra (E) bytes, defined in clauses A.3.3.3.2 and A.3.3.3.3, respectively. The first bit of each as header byte is the Header Type identification bit (H) to identify whether a header byte is a ToC or E byte. If the H bit is set to 0, the corresponding byte is a ToC byte, and if set to 1, the corresponding byte is an E byte. The second bit of a ToC byte is the Following (F) bit (see clause A.3.3.3.2), which if set to 1 indicates that another header byte is following. The last header byte shall be a ToC byte and have the F bit set to 0.

The general structure of a header byte is shown in figure A.3.3.3.1-1.

0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|H| ToC / E |
+-+-+-+-+-+-+-+-+

Figure A.3.3.3.1-1: Generic structure of a payload header byte.

H (1 bit): Header Type identification bit. For a ToC byte this is set to 0, for an E byte this is set to 1.

#### A.3.3.3.2 ToC byte

The ToC bytes define the content of the frame data in the IVAS payload following the IVAS payload header. For each IVAS or EVS frame and for each NO\_DATA frame (i.e. a frame that has zero size frame data) in the payload there shall be one ToC byte to signal the IVAS mode and bit rate. ToC bytes and the respective frame data shall be in the same order.

The Table of Content (ToC) byte structure is an extension of the ToC byte structure defined in clause A.2.2.1.2 in [3]. In the EVS payload format in [3] a code point in the ToC byte (see Figure A.5 in [3]) for extensions has been reserved, the "Unused" bit. In the present document this "Unused" bit of the Frame type index bits is activated and called "IVAS indicator" to distinguish EVS and IVAS frame data. The specific ToC structure for an IVAS frame is shown in Figure A.3.

0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|0|F|0|1| BR |
+-+-+-+-+-+-+-+-+

Figure A.3.3.3.2-1: Table of Content (ToC) byte structure for an IVAS frame.

F (1 bit): If set to 1, the bit indicates that the header byte is followed by another header byte. If set to 0, the bit indicates that this header byte is the last one in this payload and no further header bytes follows this entry.

BR (4 bits): Bit rate index as defined in Table A.1 .

Table A.1: Frame Type index when EVS mode bit = 0 and "Unused"/IVAS indicator bit = 1

|  |  |  |  |
| --- | --- | --- | --- |
| EVS/IVAS mode bit (1 bit) | IVAS indicator(1 bit) | IVAS bit rate | Indicated IVAS mode and bit rate |
| 0 | 1 | 0000 | IVAS 13.2 kbps |
| 0 | 1 | 0001 | IVAS 16.4 kbps |
| 0 | 1 | 0010 | IVAS 24.4 kbps |
| 0 | 1 | 0011 | IVAS 32 kbps |
| 0 | 1 | 0100 | IVAS 48 kbps |
| 0 | 1 | 0101 | IVAS 64 kbps |
| 0 | 1 | 0110 | IVAS 80 kbps |
| 0 | 1 | 0111 | IVAS 96 kbps |
| 0 | 1 | 1000 | IVAS 128 kbps |
| 0 | 1 | 1001 | IVAS 160 kbps |
| 0 | 1 | 1010 | IVAS 192 kbps |
| 0 | 1 | 1011 | IVAS 256 kbps |
| 0 | 1 | 1100 | IVAS 384 kbps |
| 0 | 1 | 1101 | IVAS 512 kbps |
| 0 | 1 | 1110 | Reserved[/IVAS-SR] |
| 0 | 1 | 1111 | IVAS 5.2 kbps SID |
|  |  |  |  |

[

NOTE: Split Rendering support in this payload format is under construction. This is to support the agreed ISAR feature of the IVAS codec in this RTP payload format. The reserved entry may correspond to IVAS 768 to be used when split rendering is negotiated in the session.

]

The ToC also allows signaling the EVS bit rates defined in Tables A.4 and A.5 in [3] when the EVS/IVAS mode bit is set to 1 or when the EVS/IVAS mode bit and the Unused/IVAS indicator bit are set to 0.

NO\_DATA and SPEECH\_LOST frames for both EVS and IVAS modes are signalled with the bit combinations in Table A.4 in Annex A of [3].

NOTE: Received NO\_DATA or SPEECH\_LOST frames do not relate to either EVS or IVAS modes but simply indicate a non-existent or lost frame.

[

NOTE: Split Rendering support in this payload format is under construction. The following fields are candidates that allow split rendering signalling within a split rendering session that follows the agreed ISAR feature of the IVAS codec.

Special treatment is done in case of signaling of an IVAS split rendering payload. This case is signaled with IVAS bit rate indicator = “1110” and the EVS/IVAS mode bit is set to 1 or when the EVS/IVAS mode bit and the Unused/IVAS indicator bit are set to 0. In this special case an SR-ToC byte follows unconditionally which indicates the IVAS split rendering bit rate. The structure of the SR-ToC byte shown in Figure A.3.3.3.2-2.

0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|0|U|U|U| SR-BR |
+-+-+-+-+-+-+-+-+

Figure A.3.3.3.2-2: Structure of SR-ToC byte

U (3 bits): Three unused bits. A sender shall set these bits to 0. A receiver shall ignore these bits. Note that the second left bit position of this byte shall not be interpreted as F bit.

SR-BR (4 bits): Bit rate index as defined in Table A.1a.

Table A.1a: Indicated IVAS split rendering bit rate (SR-BR)

|  |  |
| --- | --- |
| SR-BR | Indicated IVAS SR bit rate |
| 0000 | IVAS-SR 13.2 kbps |
| 0001 | IVAS-SR 16.4 kbps |
| 0010 | IVAS-SR 24.4 kbps |
| 0011 | IVAS-SR 32 kbps |
| 0100 | IVAS-SR 48 kbps |
| 0101 | IVAS-SR 64 kbps |
| 0110 | IVAS-SR 80 kbps |
| 0111 | IVAS-SR 96 kbps |
| 1000 | IVAS-SR 128 kbps |
| 1001 | IVAS-SR 160 kbps |
| 1010 | IVAS-SR 192 kbps |
| 1011 | IVAS-SR 256 kbps |
| 1100 | IVAS-SR 384 kbps |
| 1101 | IVAS-SR 512 kbps |
| 1110 | IVAS-SR 784 kbps |
| 1111 | reserved |

]

#### A.3.3.3.3 E (Extra) byte

##### A.3.3.3.3.1 General

The specific E byte structure in the IVAS payload header is shown in Figure A-4. E bytes contain extra information and shall precede the ToC bytes of the coded frames they relate to. There may be multiple E bytes preceding a ToC byte. After the initial E-byte with the CMR there may be multiple subsequent E bytes preceding ToC bytes. Subsequent E bytes may be extended by another E byte of the same type. E bytes may precede any ToC byte; E bytes in the current version of this specification are only permitted before the first ToC byte.

The E (Extra) byte structure is shown in Figure A.3.3.3.3.1-1.

0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|1| E-data |
+-+-+-+-+-+-+-+-+

Figure A.3.3.3.3.1-1: E (Extra) byte structure

Parsing of one payload header byte follows the state machine of Figure A.3.3.3.3.1-2.



Figure A.3.3.3.3.1-2: State Machine for parsing a Payload Header Byte.

##### A.3.3.3.3.2 Initial E-byte (CMR)

If a codec mode request (CMR) is sent in the current RTP packet, the initial E byte follows the structure of the CMR byte as defined in Figure A.4 of [3]. The previously "Reserved" entries of Table A.3 in 3] when the T (Type of Request) field is 111 of Figure A.4 of [3] are replaced according to Table A.1a .

Table A.1a: Structure of the CMR byte for T=111

|  |  |
| --- | --- |
| Code | Definition |
| T | D | BR |
| 111 | 0000 | IVAS 13.2 |
| 0001 | IVAS 16.4 |
| 0010 | IVAS 24.4 |
| 0011 | IVAS 32 |
| 0100 | IVAS 48 |
| 0101 | IVAS 64 |
| 0110 | IVAS 80 |
| 0111 | IVAS 96 |
| 1000 | IVAS 128 |
| 1001 | IVAS 160  |
| 1010 | IVAS 192 |
| 1011 | IVAS 256 |
| 1100 | IVAS 384 |
| 1101 | IVAS 512 |
| 1110 | [IVAS 768] |
| 1111 | NO\_REQ |
|  |

[

NOTE: The bit rate of 768 kbps shall not be used unless fragmentation is supported or an MTU size exists that does not require fragmentation. This relates to the split renderer support which is under construction.

]

CMR code-point "NO\_REQ" remains as defined in Table A.3 in [3]; it is specified as equivalent to no CMR-value being sent. The receiver of "NO\_REQ" shall ignore it.

[

NOTE: In a SR session, the CMR byte of Table A.1a is respective the first stream. Rate requests for a second non-diegetic stream are made using subsequent E-bytes.

]

The resulting byte structure is shown in Figure A.3.3.3.3.2-1.

0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|H|1|1|1| BR |
+-+-+-+-+-+-+-+-+

Figure A.3.3.3.3.2-1: Initial E byte structure for IVAS (same as EVS CMR byte structure)

BR (4 bit): IVAS bit rate as indicated in Table A.1a.

NOTE Whether t is alsorequired is ffs.

##### A.3.3.3.3.3 Subsequent E-bytes

A.3.3.3.3.3.1 General

Subsequent E byte(s) (after the initial E byte) may follow to request bandwidth (Figure A.3.3.3.3-3) or coded format (Figure A.3.3.3.3-4) or to indicate the presence of PI data (Figure A.3.3.3.3-5) in the payload. Reserved bits in the following E byte structures shall be set to 0, unless defined. The common fields in a subsequent E-byte are:

H (1 bit): Header Type identification bit. For an E byte this bit is always set to 1.

ET (2 bits): Type of subsequent E byte (00, 01, 10, 11) as indicated in Table A.1b. [The value 11 is reserved and shall not be used.]

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

A.3.3.3.3.3.2 Bandwidth Request

Bandwidth requests are defined as shown in Figure A.3.3.3.3.3.2-1 .

 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|H| ET| res | BW|
+-+-+-+-+-+-+-+-+

Figure A.3.3.3.3.3.2-1: Subsequent E byte structure for bandwidth request (ET=00)

BW (2 bits): Requested bandwidth as indicated in Table A.1c.

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

A.3.3.3.3.3.3 Coded Format Request

Coded format requests are defined as shown in Figure A.3.3.3.3.3.3-1

 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|H|ET |res| FMT |
+-+-+-+-+-+-+-+-+

Figure A.3.3.3.3.3.3-1: Subsequent E byte structure for coded format request (ET=01)

FMT (3 bits): Requested coded format as indicated in Table A.1d.

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

NOTE

A.3.3.3.3.3.4 PI Indication

PI indication to indicate PI data presence in the payload are defined as shown in Figure A.3.3.3.3.3.2-1 .

 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|H|ET | res |
+-+-+-+-+-+-+-+-+

Figure A.3.3.3.3.3.4-1: Subsequent E byte structure to indicate PI data presence in the payload (ET=10)

[

NOTE: Split Rendering support in this payload format is under construction. The following fields are candidates that allow split rendering signalling within a split rendering session that follows the agreed ISAR feature of the IVAS codec.

A.3.3.3.3.3.4x Split Renderer Request

Split renderer request are defined as shown in Figure A.3.3.3.3.3.4x-1 .

 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|H|ET |S|SC |SBR|
+-+-+-+-+-+-+-+-+

Figure A.3.3.3.3.3.4x-1: Subsequent E byte structure to structure for split renderer request (ET=11)

S (1 bit): Enable/disable split rendering as indicated in Table A.1e .

SC (2 bits): Split rendering configuration request as indicated in Table A.1f .

SBR (2 bits): Split rendering bit rate request for a second non-diegetic stream as indicated in Table A.1g .

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

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

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

Table A.1e: S bit in a subsequent E byte

|  |  |
| --- | --- |
| S | Definition |
| 0 | Disable split rendering stream associated with SC indicator |
| 1 | Enable split rendering stream associated with SC indicator |

NOTE: The S field in the table above is used for enabling or disabling a split rendering stream. The request pertains to the stream indicated by the SC field.

Table A.1f: SC field in a subsequent E byte

|  |  |
| --- | --- |
| SC | Definition |
| 00 | Stereo (0-DoF binaural or non-diegetic) |
| 01 | Diegetic with pose correction |
| 10 | Reserved |
| 11 | NO\_REQ |

NOTE: The SC field in the table above is used for split rendering configuration requests. In case, a session with a negotiated additional (non-diegetic) stream, a second E byte shall be used for split renderer request for that stream.

Table A.1g: SBR field in a subsequent E byte

|  |  |
| --- | --- |
| SBR | Definition |
| 00 | Rate option 1 |
| 01 | Rate option 2 |
| 10 | Rate option 3 |
| 11 | NQ\_REQ |

NOTE: The SBR field in the table above may be used for split rendering bit rate requests for a second non-diegetic stream.

]

## A.3.4 Frame Data

The RTP payload comprises, apart from headers, one or several coded frames, the Frame Data.

The bits are in the same order as produced by the IVAS encoder, where the first bit is placed left-most immediately following the IVAS payload header.

The format supports the transmission of EVS (primary and AMRWB-IO) and IVAS coded frames.

## A.3.5 Processing Information (PI) data

### A.3.5.1 General

In addition to one or more IVAS frame(s), one or more PI data may be transmitted as part of the PI data section in the IVAS RTP payload. PI data in the forward direction (i.e., from a media sender to a media receiver) is a mechanism to assist the processing of the IVAS frame(s) at the renderer, where the PI data is acquired prior to the rendering of IVAS frame(s). [PI data in the reverse direction (i.e., from a media receiver to a media sender) includes sending requests and feedback to control a media sender. Consequently, PI data provides bi-directional signalling between the media sender and the media receiver. ]

NOTE: When EVS is used for one call leg that the usage of the the IVAS PLF for enabling the PI data concept is ffs.

Figure A.3.5.1-1 presents the structure of a PI data section which is formed by the PI-specific header section followed by the PI frame data. The PI header section (see clause A.3.5.a) identifies the different PI data frames (see clause A.3.5x2). The PI data section is included in the IVAS RTP payload as described in clause A.3.3.2 and in figure A.1.

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | PI header section | PI frame data |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 \---------------PI data---------------/

**Figure A.3.5.1-1: The structure of a PI data section.**

### A.3.5.2 PI data header

As presented in clause A.3.5.1 and in figure A.3.5.1-1, a PI data block contains a header section and a frame data section. Each PI header identifies the type and size for the associated PI data frame. Furthermore, the PI header identifies to which audio frame(s) the PI data is associated with through the PI frame marker bits (PM). All the PI header indications are added to the PI header section in a row before the PI data frames.

PI data header indication structure is presented in figure A.4f. The header elements are defined as:

PF (1 bit): If set to 1, the bit indicates that the PI header indication is followed by another PI header indication. If set to 0, the bit indicates that this PI header indication is the last one in this payload and no further PI header indication follows this entry.

PM (2 bits): PI frame marker bits indicate the associated audio frame for the PI data frame, see Table A.1e.

PI type (5 bits): PI type bits indicate the type for the PI data.

PI size (8 bits): PI size bits indicate the size for the PI data frame in bytes, see table A.1x2. A size of zero indicates that there is no PI data frame associated for the PI header indication.

0 1 \_
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
 +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 |PF| PM | PI type | PI size |
 +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

Figure A.4f: PI data header.

Table A.1x1: PF bits for a PI header.

|  |  |
| --- | --- |
| **PF bits** | **Indication** |
| 0 | Last header indication |
| 1 | Header indication follows |

Table A.1e: Marker bits for a PI header byte.

|  |  |
| --- | --- |
| PM bits | Frame marker indication |
| 00 | Reserved |
| 01 | Not last PI header for this frame |
| 10 | Last PI header for this frame |
| 11 | General (applied to all frames) |

The PI headers are listed in the beginning of a PI data section in order. First, the PI headers with marker bits PM=11 (i.e., the generally applied PI data is identified first). The following PI headers are associated with the audio frames in the payload in order. PM=01 indicates that the current PI header/data is not the last one for the current audio frame and more PI headers/data will follow. PM=10 indicates that this PI header/data is the last one that is associated with the current audio frame. The next PI data is then associated with the next audio frame, i.e. the time stamp of the PI frame is increase by 320 ticks. The parsing of a PI header indication is illustrated in a state machine in Figure A.3.5.2-1.



Figure A.3.5.2-1: State machine for parsing a PI header byte.

Table A.1x2 indicates the values for the “size” bits in the PI header. A size value of zero indicates that there is no associated PI data for this header entry. A value of (1111 1111) indicates that another size byte follows. The size of the PI data is then calculated as the sum of the respective sizes indicated by the “size” bytes, where a size of 255 bytes is used for the “size” byte with (1111 1111) bit sequence. For example, if there are two “size” bytes, a first byte of (1111 1111) and a second byte of (0000 1111, indicating a size of 15 bytes), the size of the PI data is 255 + 15 = 270 bytes. Recursive size indication with subsequent “size” bytes is also possible.

Table A.1x2: Size bits for PI header.

|  |  |
| --- | --- |
| **PI Size bits** | **Indicated size in bytes (or other indication)** |
| 0000 0000 | 0 |
| 0000 0001 | 1 |
| … | … |
| 1111 1110 | 254 |
| 1111 1111 | Another size byte follows |

### A.3.5.3 Media time when IVAS PI data is included in RTP packets

When the IVAS codec is used, then RTP packets may include both PI data and audio data, and the PI data may need to be synchronized with the audio data.

NOTE: Sending PI data in RTCP and/or in RTP header extension is ffs.

When forward direction PI data is included in the RTP packets, the following applies:

- The PI data is associated with an audio frame and use the media time of the audio data.

- If PI data needs to be transmitted and no audio frame is available, e.g., during DTX periods, then a NO\_DATA frame is included in the packet containing the PI data. Alternatively, the PI data can be transmitted with the SID frames. See clause A.3.5.yy for more information.

- If PI data is not needed to be transmitted when there is no audio frame available, e.g., during DTX periods, then the transmission of the PI data can be delayed until an audio frame is available. If there are multiple PI data frames with the same type available after a delay period, the most recent PI data may be selected for transmission (e.g., the most recent device orientation may be transmitted). The other (older) PI data frames with the same type may be ignored. Depending on the type of the delayed PI data frames, all of the PI data frames with the same type may be transmitted. See clause A.3.5.yy for more information.

- When receiving an RTP packet with both PI data and several audio frames:

- the media time of the first audio frame is calculated from the RTP time stamp,

- the media time of a subsequent audio frames is calculated from the media time of a preceding audio frame by adding 20 ms.

- PI data does not add to the media time.

- PI data can be sent in a separate RTP packet from the audio frame and then use the media time calculated from the RTP time stamp.

 [

When reverse direction PI data is included in the RTP packets, the following applies:

- The PI data is associated with the media time of the received audio data to which such reverse data is applied

- An explicit timestamp is provided for PI data that is a sampled value (e.g. orientation data)

- PI data that is a pure request does note have a sampling instant and would be interpreted by the recording device on a best-effort basis

When receiving reverse direction PI data in RTP packets, the receiving device may synchronize PI data with a timestamp to its sender media timeline and use this correspondence as the basis for applying orientations.

]

NOTE: It is FFS if more rules are needed.

### A.3.5.4 PI data handling during DTX

During DTX periods, when determined, no audio frames are transmitted. Consequently, there is no temporal parameter, i.e., RTP time stamp, available that would be associated with the media time of the audio which would furthermore be associated with the PI data. Therefore, if PI data is obtained during a DTX period, the transmission of the PI data is controlled by other means. If the PI data needs to be transmitted as soon as possible, i.e., during the DTX period, the PI data can be associated with the next transmitted SID frame. Alternatively, the PI data can be transmitted immediately with an associated NO\_DATA frame in the payload. In these cases, the RTP time stamp for the PI data transmission is obtained from the media time of the respective SID or NO\_DATA frames.

If the transmission of the PI data can wait until the DTX period is over, the transmission of the PI data can be delayed until the next audio frame is available. In this case, the RTP time stamp for the PI data transmission is obtained from the media time of the respective next audio frame.

It is possible that multiple PI data with the same type are obtained during the DTX period. For example, multiple device orientation PI data frames can be obtained during the DTX period. If the transmission of the PI data is delayed, there can be multiple PI data frames (e.g., of device orientation type) waiting to be transmitted when the DTX period ends. In this case, the latest PI data obtained during the DTX period can be selected to be transmitted (e.g., the latest device orientation is selected to be transmitted). Alternatively, all of the delayed PI data may be transmitted after the DTX period ends. For example, if multiple acoustic environment PI data are delayed, it is beneficial to transmit them all instead of just the latest one. In this case, all of the delayed PI data can be associated to the first audio frame after the DTX period or to mark the PI data with PM=11.

### A.3.5.4 Supported PI data types

Supported PI types are listed in tables A.1x4[, A.1x5 and A.1x6] and described in the following subsections. Table A.3.5.4-1 lists PI types for forward direction signalling [and table A.3.5.4x1-1 lists PI types for reverse direction signalling. Table A.3.5.4x2-1 lists additional PI types].

**Table A.3.5.4-1 : Supported forward direction PI types in an IVAS session.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type bits** | **Forward direction PI type** | **Description** | **SDP indication** | **Size (bytes)** | **Described in clause** |
| 00000 | SCENE\_ORIENTATION | Describes the orientation of a spatial audio scene either in Quaternions, Euler angles or spherical coordinates. | fsco | 8 | A.3.5x2.1.1.2 |
| 00001 | DEVICE\_ORIENTATION\_COMPENSATED | Describes the orientation of a device either in Quaternions or Euler angles. The orientation is compensated in the transmitted audio. | fdoc | 8 | A.3.5x2.1.1.3 |
| 00010 | DEVICE\_ORIENTATION\_UNCOMPENSATED | Describes the orientation of a device either in Quaternions or Euler angles. The orientation is not compensated in the transmitted audio. | fdou | 8 | A.3.5x2.1.1.3 |
| 00011 | ACOUSTIC\_ENVIRONMENT | Selects and optionally describes the acoustic environment. | face | 1 or 5 | A.3.5x2.1.2 |
| [ |  |  |  |  |  |
| 00100 | DISABLE\_HEADTRACKING | Disables headtracking at the receiver. | fdht | 0 | A.3.5x2.1.3 |
| 00101 | ENABLE\_HEADTRACKING | Enables headtracking at the receiver. | feht | 0 | A.3.5x2.1.3 |
| 00110 | N\_ISM | Number of ISMs. Also includes the number of active ISMs. | fnis | 1 | A.3.5x2.1.4.3 |
| 00111 | ISM\_DISTANCE\_ATTENUATION | Describes distance attenuation for all ISMs. | fida | 3 | A.3.5x2.1.4.4 |
| 01000 | ISM\_ID | Indicates ID of each transported ISM. | fiid | 1 + N ISM x 1 | A.3.5x2.1.4.5 |
| 01001 | ISM\_GAIN | Describes gain factor for each ISM. | figa | 1 + N ISM x 1 | A.3.5x2.1.4.6 |
| 01010 | ISM\_ORIENTATION | Describes an orientation for each ISM. | fiso | 1 + N ISM x 8 | A.3.5x2.1.4.7 |
| 01011 | ISM\_POSITION | Describes a position of each ISM. | fiso | 1 + N ISM x 6 | A.3.5x2.1.4.8 |
| 01100 | ISM\_EXTENT | Describes an extent/spread for each ISM. | fiex | 1 + N ISM x ? | A.3.5x2.1.4.9 |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 01101 | ISM\_DIRECTIVITY | Describes directivity for each ISM. | fidr | 1 + N ISM x 2 | A.3.5x2.1.4.10 |
| 01110 | ISM\_PANNING | Describes panning for each ISM. | fipa | 1 + N ISM x 1 | A.3.5x2.1.4.11 |
| 01111 | ISM\_PROXIMITY | Describes proximity between ISMs. | fipr | 2-7 | A.3.5x2.1.4.12 |
| ] |  |  |  |  |  |

[

**Table A.1x5 : Supported reverse direction PI types in an IVAS session.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type bits** | **Reverse direction PI type** | **Description** | **SDP indication** | **Size (bytes)** | **Described in clause** |
| 10000 | PLAYBACK\_DEVICE\_ORIENTATION | Feedback. Describes the orientation of the playback device either in Quaternions, Euler angles or spherical coordinates. Split rendering related.  | rpdo | [4+]8 | A.3.5x2.2.1.2 |
| 10001 | HEAD\_ORIENTATION | Feedback. Describes the head orientation of the listener either in Quaternions, Euler angles or spherical coordinates. Split rendering related. | rhor | [4+]8 | A.3.5x2.2.1.3 |
| 10010 | LISTENER\_POSITION | Feedback. Describes the position of the listener in 3D space. Split rendering related. | rlip | [4+]6 | A.3.5x2.2.2 |
| 10011 | DISABLE\_DEVICE\_ORIENTATION\_COMPENSATION | Request. A request to disable orientation compensations applied at the sender side. | rdoc | 0 | A.3.5x2.2.3 |
| 10101 | ENABLE\_DEVICE\_ORIENTATION\_COMPENSATION | Request. A request to enable orientation compensations at the sender side. | reoc | 0 | A.3.5x2.2.3 |
| 10110 | LATENCY | Feedback. Describes the latency of other PI feedback data until it is applied. | rlat | 4 | A.3.5x2.2.4 |

**Table A.1x6 : Additional PI types in an IVAS session.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type bits** | **PI type** | **Description** | **SDP indication** | **Size (bytes)** |
| 10111-11110 | Reserved | - | - | - |
|  |  |  |  |  |
| 11111 | NO\_PI\_DATA | Indicates an empty PI data frame. | nopi | 0 |

NO\_PI\_DATA PI data type can be used to indicate empty PI data sections. The PM marker bits for a NO\_PI\_DATA PI data type shall be set as PM=10, see table A.1e. For example, if an IVAS RTP payload includes multiple audio frames, and some of the audio frames do not have associated PI data, NO\_PI\_DATA PI type can be used.

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

]

### A.3.5.5 Forward direction PI data types

#### A.3.5.5.1 Orientation PI data (forward direction)

##### A.3.5.5.1.1 Orientation data structures

Figures A.1f3.1 and A.1f3.2 below show PI orientation data structures in quaternion and Euler angles conventions. Figure A.1f3.1 shows orientation data structure as quaternions and each angle has 16 bits reserved for the value. The quaternion component values range from -1 to 1 as stated in. With 16 bits values, the resolution for a single component is . The represented quaternion is a unit quaternion. Following the IVAS coordinate system in 7.4.3.1, a quaternion of (w=0, x=1, y=0, z=0) represents the frontal direction. The positive x-axis points towards the frontal direction, the positive y-axis points towards the left direction and the positive z-axis points towards the up direction.

 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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | W | X |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Y | Z |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

**Figure A.1f3.1: PI orientation data as quaternions.**

Figure A.1f3.2 shows how Euler angles (yaw, pitch, roll) can be transmitted in an orientation PI data frame. The first value (two bytes in this example) indicates a value of -3 (which cannot occur for unit quaternions) which indicates that the rest of the orientation data should be interpreted as Euler angles. In figure A.1f3.2, the orientation data structure has 16 bits reserved for each angle value. The angles range from -180° (exclusive) to 180°. With 16 bits values, the angle resolution is . Yaw indicates a right-hand positive orientation around the z-axis (up direction), pitch indicates a left-hand positive orientation around the y-axis (left direction) and roll indicates left-hand positive orientation around the x-axis (front direction), see clause 7.4 for more information on the IVAS coordinate system.

 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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | -3 | yaw |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | pitch | roll |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

**Figure A.1f3.2: PI orientation data as Euler angles.**

The received orientations can be transmitted to the external orientation handling and processed as stated in clause 7.4.4.

##### A.3.5.5.1.2 Scene orientation

The SCENE\_ORIENTATION PI data describes the orientation of the capturer side scene. The frontal direction of the scene points towards the capture frontal direction. See also clause 7.4.2.1 (Scene orientation). In the PI data, the azimuth and elevation values of the scene orientation are transformed into (unit) quaternions. A frontal direction of zero azimuth and zero elevation corresponds to (w=0, x=1, y=0, z=0) in quaternions, see clause 7.4.3.1 for the IVAS coordinate system.

The SCENE\_ORIENTATION PI data is applied to the audio frame with the same timestamp. The latest received SCENE\_ORIENTATION PI data is used until a new SCENE\_ORIENTATION PI data is received.

##### A.3.5.5.1.3 Device orientation

The DEVICE\_ORIENTATION\_COMPENSATED and DEVICE\_ORIENTATION\_UNCOMPENSATED PI data describes the orientation of the sender (capture) device. I.e., the orientation indicates the orientation deviation from the frontal capture direction, when the device is used for audio capture. The frontal direction of the device point towards the capture frontal direction, i.e. (w=0, x=1, y=0, z=0) in quaternions, see clause 7.4.3.1 for the IVAS coordinate system.

The orientation of a capturing device can be used to stabilize the captured spatial audio content by, e.g., removing undesirable orientation changes. The sender device may be a non-stationary capturing device (e.g., a mobile phone with multiple microphones or a spatial audio capture microphone array). For example, a caller is holding a mobile phone in their hand or wearing a head-mounted display with spatial audio capturing capabilities. Some of the movements of the caller that affect the spatial audio capture can be undesirable (e.g., hand or head movements). The undesirable movements (or orientations) can be compensated in the transmitted spatial audio content via the DEVICE\_ORIENTATION\_COMPENSATED PI type. When the device orientation is compensated in the transmitted spatial audio content, the received spatial audio can be rendered to the receiver (in their user space) without experiencing the undesirable orientations or movements. In case the spatial audio content is transmitted without orientation compensation, the compensation can be performed by the receiver device. In this case, DEVICE\_ORIENTATION\_UNCOMPENSATED PI type can be used to transmit the change in the capture device orientation.

DEVICE\_ORIENTATION\_COMPENSATED PI data indicates that the transmitted orientation is already compensated in the related transmitted audio.

DEVICE\_ORIENTATION\_UNCOMPENSATED PI data indicates that the transmitted orientation is not compensated in the related transmitted audio.

The device orientation PI data is applied to the audio frame with the same timestamp. The latest received device orientation PI data is used until a new device orientation PI data is received.

#### A.3.5.5.2 Acoustic environment PI data

Acoustic environment (AE) PI data frames can be used to transmit room acoustic data. The room acoustic data consist of late reverb parameters and optionally early reflections parameters. Late reverb parameters include:

- RT60 – indicating the time that it takes for the reflections to reduce 60 dB in energy level, per frequency band,

- DSR – diffuse to source signal energy ratio, per frequency band,

- Pre-delay – delay at which the computation of DSR values was performed, which can be also seen as the threshold point between early reflections and late reverberation phase.

Both RT60 and DSR parameters are specified per frequency band. Pre-defined or custom frequency bands can be used. Pre-delay is a scalar.

The early reflections parameters include:

- 3D rectangular virtual room dimensions,

- Broadband energy absorption coefficient per wall,

- Listener origin coordinates (optional),

- Low-complexity mode flag (optional).

See clause 7.4.8 (Room acoustics parameters) for more detailed description of room acoustics parameters.

The acoustic environment data can be provided upfront using PI frame in SDP header, using base64 coding. In such case the payload shall be provided according to the format described in TS 26.258, Annex B regarding payloadAcEnv syntax element. Note, that payloadAcEnv supports multiple acoustic environments using different frequency grid representations. Once available, acoustic environment can be selected by using its identifier.

To control acoustic environments runtime, acoustic environment PI data frames can be used. An acoustic environment PI data frame can contain:

- an acoustic environment identifier alone (7 bits),

- a compact representation of the acoustic environment (40 bits),

The content of the PI frame received is determined by its size.

Acoustic environment can be selected by sending an acoustic environment PI frame containing a seven-bit AE identifier, as illustrated in figure A.1f4.1. Such an acoustic environment should be available upfront and provided either using SDP header or another AE PI data frame containing either full or compact acoustic environment representation. It allows for the following graceful degradation mechanism:

- full AE representation should be used if available, otherwise:

- compact AE representation should be used if available, otherwise:

- default AE definition should be used if none of the above are available.

 0 1 2 3 4 5 6 7
 +-+-+-+-+-+-+-+-+
 |0| ID |
 +-+-+-+-+-+-+-+-+

Figure A.1f4.1: Acoustic environment PI data frame (ACOUSTIC\_ENVIRONMENT) containing an AE identifier.

Acoustic environment data can also get updated real-time. In case of real-time updates, the AE data can be transmitted as compact packets using RTP protocol.

NOTE: The transmission mechanism for complete payloadAcEnv syntax elements is ffs.

Compact packets contain a low-resolution representation of the acoustic environment. This facilitates avoiding spikes in transmission and is better suited for repeated transmission allowing for instant synthesis of the new acoustic environment, although with initially reduced accuracy. A compact acoustic environment PI data frame is presented in figure A.1f4.2. The data frame consists of an acoustic environment ID field (7 bits), three RT60 fields (5 bits each) and three DSR fields (6 bits each) summing up to 40 bits size. The RT60 and DSR values are provided for three frequency bands of center frequency {25 Hz, 250 Hz, 2.5 kHz}, which are denoted {lo, mi, hi} in figure A.1f4.2 respectively. The RT60 fields use 5-bit code *n* representing duration in seconds according to formula with . The 5-bit codes and related RT60 durations are also shown in table A.1x7.1.

Table A.1x7.1 : 5-bit codes and respective RT60 values

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Code | Value | Code | Value | Code | Value | Code | Value |
| 00000 | 0.01 | 01000 | 0.0635 | 10000 | 0.4032 | 11000 | 2.56 |
| 00001 | 0.0126 | 01001 | 0.08 | 10001 | 0.5080 | 11001 | 3.2254 |
| 00010 | 0.0159 | 01010 | 0.1008 | 10010 | 0.64 | 11010 | 4.0637 |
| 00011 | 0.02 | 01011 | 0.1270 | 10011 | 0.8063 | 11011 | 5.12 |
| 00100 | 0.0252 | 01100 | 0.16 | 10100 | 1.0159 | 11100 | 6.4508 |
| 00101 | 0.0317 | 01101 | 0.2016 | 10101 | 1.28 | 11101 | 8.1275 |
| 00110 | 0.04 | 01110 | 0.2540 | 10110 | 1.6127 | 11110 | 10.24 |
| 00111 | 0.0504 | 01111 | 0.32 | 10111 | 2.0319 | 11111 | 12.9016 |

The DSR values are computed as [dB] with resulting in the range between ‑20 and ‑83 dB.

Table A.1x7.2 : 6-bit codes and respective DSR values

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Code | Value | Code | Value | Code | Value | Code | Value |
| 000000 | -20 | 010000 | -36 | 100000 | -52 | 110000 | -68 |
| 000001 | -21 | 010001 | -37 | 100001 | -53 | 110001 | -69 |
| 000010 | -22 | 010010 | -38 | 100010 | -54 | 110010 | -70 |
| 000011 | -23 | 010011 | -39 | 100011 | -55 | 110011 | -71 |
| 000100 | -24 | 010100 | -40 | 100100 | -56 | 110100 | -72 |
| 000101 | -25 | 010101 | -41 | 100101 | -57 | 110101 | -73 |
| 000110 | -26 | 010110 | -42 | 100110 | -58 | 110110 | -74 |
| 000111 | -27 | 010111 | -43 | 100111 | -59 | 110111 | -75 |
| 001000 | -28 | 011000 | -44 | 101000 | -60 | 111000 | -76 |
| 001001 | -29 | 011001 | -45 | 101001 | -61 | 111001 | -77 |
| 001010 | -30 | 011010 | -46 | 101010 | -62 | 111010 | -78 |
| 001011 | -31 | 011011 | -47 | 101011 | -63 | 111011 | -79 |
| 001100 | -32 | 011100 | -48 | 101100 | -64 | 111100 | -80 |
| 001101 | -33 | 011101 | -49 | 101101 | -65 | 111101 | -81 |
| 001110 | -34 | 011110 | -50 | 101110 | -66 | 111110 | -82 |
| 001111 | -35 | 011111 | -51 | 101111 | -67 | 111111 | -83 |

No pre-delay value is transmitted. It gets computed as one tenth of RT60 of 250 Hz band.

Once a compact packet is received, an acoustic environment should be switched to the one specified with its ID. If a full representation is available already, it should prevail over the compact representation. To facilitate minimizing delay and improving reliability, compact acoustic environment representation should be repeated with relevant frequency, e.g., every couple of seconds if needed.

 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 | RT60 lo | DSR lo | RT60 mi | DSR mi | RT60
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 hi | DSR hi |
 +-+-+-+-+-+-+-+-+

Figure A.1f4.2: Acoustic environment PI data frame (ACOUSTIC\_ENVIRONMENT) containing a compact AE representation.

#### A.3.5.5.x1 Disable and enable headtracking

DISABLE\_HEADTRACKING PI frame indicates that the headtracking should not be used for the transmitted audio. I.e., the head orientation of the listener should not be applied when rendering the transmitted audio. When disabling the headtracking, the head orientation should be reset to a default pose corresponding to (w=0, x=1, y=0, z=0) in quaternions, see clauses A.3.5x2.2.1.3 and 7.4.2.2 (Listener orientation). No extra PI data section is transmitted with DISABLE\_HEADTRACKING PI data (i.e., the PI data frame section is empty), the indication for disabling head tracking is recognised from the PI type alone.

NOTE: How fast this has to happen is ffs.

DISABLE\_HEADTRACKING PI data disables the headtracking for the current audio frame and for any future audio frames. The headtracking can be re-enabled by sending ENABLE\_HEADTRACKING PI frame to the receiver. These PI data are recommendations from the sender device and the receiver device may choose to override these recommendations.

The sender device can provide indication to the receiver device to disable headtracking if there is a need to ensure that the spatial audio content is consumed without responding to head movements and to lock the head orientation to a default pose. For example, DISABLE\_HEADTRACKING PI data created by the sender can be transmitted to the receiver device, if the transmitted spatial audio content already includes orientation changes and no further orientation changes are desired from the headtracking. DISABLE\_HEADTRACKING PI data may also be used to lock a source to a certain orientation irrespective to the head orientation (e.g., for an audio notification or alert).

#### A.3.5.5.x2 ISM specific PI data

##### A.3.5.5.x2.1 ISM specific PI data types and structure

ISM related PI data types are listed in table A.1x12. Figure A.1f18 presents a general PI data structure for ISM related PI types. The data structure begins with an ISM specific PI header (see clause A.3.5x2.1.4.2) followed by the PI data frames for each ISM (the number of data frames depends on the number of ISMs).

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ISM PI header | ISM1 PI data | ISM2 PI data |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ISM3 PI data | ISM4 PI data |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

**Figure A.1f18: ISM PI data structure with 4 ISM objects.**

**Table A.1x12 : ISM related PI data types.**

|  |
| --- |
| **ISM related PI type** |
| N\_ISM |
| ISM\_DISTANCE\_ATTENUATION |
| ISM\_ID |
| ISM\_GAIN |
| ISM\_ORIENTATION |
| ISM\_POSITION |
| ISM\_EXTENT |
|  |
| ISM\_DIRECTIVITY |
| ISM\_PANNING |
| ISM\_PROXIMITY |

##### A.3.5.5.x2.2 ISM specific PI header

Each ISM specific PI data frame is preceded by a header indicating the number of ISMs in the associated IVAS frame and a diegetic/non-diegetic indicator for each ISM. In the ISM header in figure A.1f19, the first two bits indicate the number of ISMs (N) according to table A.1x13. The next four bits act as individual flags for each ISM starting from the first ISM. Each flag (X) indicates if the audio object is non-diegetic (1) or not (0). The last two bits are used for zero padding to force byte alignment.

0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
| N |X|X|X|X|0 0|
+-+-+-+-+-+-+-+-+

**Figure A.1f19: ISM header indicating the number of ISMs and flags to indicate non-diegetic ISMs.**

**Table A.1x13 : N bit values in the ISM header and the indicated number of ISMs.**

|  |  |
| --- | --- |
| **N bits** | **Indicated number of ISMs** |
| 00 | 1 |
| 01 | 2 |
| 10 | 3 |
| 11 | 4 |

##### A.3.5.5.x2.3 Number of ISMs

N\_ISM PI type can be used to signal the number of ISMs in the associated IVAS frame. The N\_ISM PI data also includes the number of active ISMs for the frame. The PI data structure to signal the number of ISMs and active ISMs is the same as presented in clause A.3.5x2.1.4.2 and in figure A.1f19. I.e., the N\_ISM data only contains the ISM specific PI header.

NOTE: Whether this is this needed if there is the ISM specific PI header is ffs.

##### A.3.5.5.x2.4 ISM distance attenuation

The ISM\_DISTANCE\_ATTENUATION PI data frame is presented in figure A.1f20. The data frame includes a reference distance (6 bits), maximum distance (6 bits) and a roll-off factor (6 bits), totalling in 3 bytes of data (including zero padding for byte alignment), see table XX1-XX3. ISM distance attenuation is a rendering configuration applicable for all rendered ISMs. For more information about ISM distance attenuation, see clause 7.2.2.2.7.

0 1 2 \_
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Ref dist. | Max dist. | Roll-off |0 0 0 0 0 0|
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

**Figure A.1f20: ISM\_DISTANCE\_ATTENUATION PI data frame.**

Table XX1: 6-bit codes and respective Reference distance values (m)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Code | Value | Code | Value | Code | Value | Code | Value |
| 000000 | 0.1 | 010000 | 1.7 | 100000 | 3.3 | 110000 | 4.9 |
| 000001 | 0.2 | 010001 | 1.8 | 100001 | 3.4 | 110001 | 5.0 |
| 000010 | 0.3 | 010010 | 1.9 | 100010 | 3.5 | 110010 | 5.1 |
| 000011 | 0.4 | 010011 | 2.0 | 100011 | 3.6 | 110011 | 5.2 |
| 000100 | 0.5 | 010100 | 2.1 | 100100 | 3.7 | 110100 | 5.3 |
| 000101 | 0.6 | 010101 | 2.2 | 100101 | 3.8 | 110101 | 5.4 |
| 000110 | 0.7 | 010110 | 2.3 | 100110 | 3.9 | 110110 | 5.5 |
| 000111 | 0.8 | 010111 | 2.4 | 100111 | 4.0 | 110111 | 5.6 |
| 001000 | 0.9 | 011000 | 2.5 | 101000 | 4.1 | 111000 | 5.7 |
| 001001 | 1.0 | 011001 | 2.6 | 101001 | 4.2 | 111001 | 5.8 |
| 001010 | 1.1 | 011010 | 2.7 | 101010 | 4.3 | 111010 | 5.9 |
| 001011 | 1.2 | 011011 | 2.8 | 101011 | 4.4 | 111011 | 6.0 |
| 001100 | 1.3 | 011100 | 2.9 | 101100 | 4.5 | 111100 | 6.1 |
| 001101 | 1.4 | 011101 | 3.0 | 101101 | 4.6 | 111101 | 6.2 |
| 001110 | 1.5 | 011110 | 3.1 | 101110 | 4.7 | 111110 | 6.3 |
| 001111 | 1.6 | 011111 | 3.2 | 101111 | 4.8 | 111111 | 6.4 |

Table XX2: 6-bit codes and respective Maximum distance values (m)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Code | Value | Code | Value | Code | Value | Code | Value |
| 000000 | 1 | 010000 | 17 | 100000 | 33 | 110000 | 49 |
| 000001 | 2 | 010001 | 18 | 100001 | 34 | 110001 | 50 |
| 000010 | 3 | 010010 | 19 | 100010 | 35 | 110010 | 51 |
| 000011 | 4 | 010011 | 20 | 100011 | 36 | 110011 | 52 |
| 000100 | 5 | 010100 | 21 | 100100 | 37 | 110100 | 53 |
| 000101 | 6 | 010101 | 22 | 100101 | 38 | 110101 | 54 |
| 000110 | 7 | 010110 | 23 | 100110 | 39 | 110110 | 55 |
| 000111 | 8 | 010111 | 24 | 100111 | 40 | 110111 | 56 |
| 001000 | 9 | 011000 | 25 | 101000 | 41 | 111000 | 57 |
| 001001 | 10 | 011001 | 26 | 101001 | 42 | 111001 | 58 |
| 001010 | 11 | 011010 | 27 | 101010 | 43 | 111010 | 59 |
| 001011 | 12 | 011011 | 28 | 101011 | 44 | 111011 | 60 |
| 001100 | 13 | 011100 | 29 | 101100 | 45 | 111100 | 61 |
| 001101 | 14 | 011101 | 30 | 101101 | 46 | 111101 | 62 |
| 001110 | 15 | 011110 | 31 | 101110 | 47 | 111110 | 63 |
| 001111 | 16 | 011111 | 32 | 101111 | 48 | 111111 | 64 |

Table XX3: 6-bit codes and respective Roll-off factor values

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Code | Value | Code | Value | Code | Value |
| 000000 | 0 | 010000 | 1.6 | 100000 | 3.2 |
| 000001 | 0.1 | 010001 | 1.7 | 100001 | 3.3 |
| 000010 | 0.2 | 010010 | 1.8 | 100010 | 3.4 |
| 000011 | 0.3 | 010011 | 1.9 | 100011 | 3.5 |
| 000100 | 0.4 | 010100 | 2.0 | 100100 | 3.6 |
| 000101 | 0.5 | 010101 | 2.1 | 100101 | 3.7 |
| 000110 | 0.6 | 010110 | 2.2 | 100110 | 3.8 |
| 000111 | 0.7 | 010111 | 2.3 | 100111 | 3.9 |
| 001000 | 0.8 | 011000 | 2.4 | 101000 | 4.0 |
| 001001 | 0.9 | 011001 | 2.5 | 101001-111111 | reserved |
| 001010 | 1.0 | 011010 | 2.6 |  |  |
| 001011 | 1.1 | 011011 | 2.7 |  |  |
| 001100 | 1.2 | 011100 | 2.8 |  |  |
| 001101 | 1.3 | 011101 | 2.9 |  |  |
| 001110 | 1.4 | 011110 | 3.0 |  |  |
| 001111 | 1.5 | 011111 | 3.1 |  |  |

##### A.3.5.5.x2.5 ISM ID

ISM\_ID PI data specifies an identity (ID) for the ISMs transported in the associated IVAS frame, as presented in figure XX. The PI data includes an identity field (one byte) for each transported ISM, positioned after one another. For example, the ISM ID for the first object is followed by the ISM ID for the second object when the number of ISMs N>1.

0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
| ISM ID |
+-+-+-+-+-+-+-+-+

**Figure XX: ISM\_ID PI data frame.**

##### A.3.5.5.x2.6 ISM gain

ISM\_GAIN PI data specifies a gain factor for ISMs transported in the associated IVAS frame, as presented in figure XX. The PI data includes ISM gain field (one byte) for each transported ISM, positioned after one another. For example, the ISM gain for the first object is followed by the ISM gain for the second object when the number of ISMs N>1. The 8-bit ISM gains represent a uniform range (in dB) ranging from -96 dB to +3 dB with additional code point for muting (-Inf dB), according to table XX.

0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
| ISM gain |0|
+-+-+-+-+-+-+-+-+

**Figure XX: ISM\_GAIN PI data frame.**

Table XX: 7-bit codes and respective gain values (dB)

|  |  |  |  |
| --- | --- | --- | --- |
| Code | Value | Code | Value |
| 0000000 | 0 |  |  |
| 0000001 | -1 | 1110001 | -87 |
| 0000010 | -2 | 1110010 | -88 |
| 0000011 | -3 | 1110011 | -89 |
| 0000100 | -4 | 1110100 | -90 |
| 0000101 | -5 | 1110101 | -91 |
| 0000110 | -6 | 1110110 | -92 |
| 0000111 | -7 | 1110111 | -93 |
| 0001000 | -8 | 1111000 | -94 |
| 0001001 | -9 | 1111001 | -95 |
| 0001010 | -10 | 1111010 | -96 |
| 0001011 | -11 | 1111011 | -Inf |
| 0001100 | -12 | 1111100 | +1 |
| 0001101 | -13 | 1111101 | +2 |
| 0001110 | -14 | 1100100 | +3 |
|  |  | 1100100-1111111 | reserved |

##### A.3.5.5.x2.7 ISM orientation

ISM\_ORIENTATION PI data describes the orientation of audio object(s) (ISMs) by orientation data structures in accordance with clause A.3.5x2.1.1.1, with respect to the scene orientation. The ISM specific header indicates the number of orientations (i.e., the number of objects) in the PI orientation data section. In the orientation data section, the full orientation representation for each audio object is positioned after one another. For example, the orientation data section begins with the quaternion components (w,x,y,z) for the first object, followed by the components for the second object and so on.

Editor's Note: IVAS renderer is probable limited to Euler angles. Should we allow quaternions still?

##### A.3.5.5.x2.8 ISM position

ISM\_POSITION PI frame indicates the position(s) of audio object(s) in 3D space. The number of position values (i.e., the number of objects) is indicated in the ISM PI header, see clause A.3.5x2.1.4.2. Figure A.1f15 shows a general position PI data structure as cartesian coordinates (X, Y, Z) that can be used for ISM\_POSITION PI data frames. Figure A.1f16 a general position PI data structure as spherical coordinates (azimuth, elevation, radius) that can also be used for ISM\_POSITION PI data frames. See clause A.3.5x2.2.2 for more information about position PI data.

##### A.3.5.5.x2.9 ISM extent

NOTE: ISM extent is ffs.

##### A.3.5.5.x2.10 ISM directivity

ISM\_DIRECTIVITY PI data frame structure is presented in figure A.1f21. The ISM orientation indicate the direction (of the inner cone) for an ISM object. The inner cone angle, “Inner ang”, (5 bits) determines the size of the main cone directed to the front direction of the object, see table XX. The outer cone angle, “Outer ang”, (5 bits) determines the size of the outer (back) cone, see table XX. The gain for the inner cone is determined by the ISM gain, and the outer attenuation gain, “Outer att”, (5 bits) determines the attenuation outside the outer cone, see table XX. The total size of an ISM\_DIRECTIVITY PI data frame for a single object is 2 bytes, including zero-padding for byte alignment. For more information about ISM directivity, see clause 7.2.2.2.7.

0 1 \_
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |Inner ang|Outer ang|Outer att|0|
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

**Figure A.1f21: ISM\_DIRECTIVITY PI data frame.**

Table A.1x7.2: 5-bit codes and respective Inner or Outer cone angle values (deg)

|  |  |  |  |
| --- | --- | --- | --- |
| Code | Value | Code | Value |
| 00000 | 0 | 10000 | 240 |
| 00001 | 15 | 10001 | 255 |
| 00010 | 30 | 10010 | 270 |
| 00011 | 45 | 10011 | 285 |
| 00100 | 60 | 10100 | 300 |
| 00101 | 75 | 10101 | 315 |
| 00110 | 90 | 10110 | 330 |
| 00111 | 105 | 10111 | 345 |
| 01000 | 120 | 11000 | 360 |
| 01001 | 135 | 11001-11111 | reserved |
| 01010 | 150 |  |  |
| 01011 | 165 |  |  |
| 01100 | 180 |  |  |
| 01101 | 195 |  |  |
| 01110 | 210 |  |  |
| 01111 | 225 |  |  |

Table A.1x7.2: 5-bit codes and respective Outer attenuation gain values (dB)

|  |  |  |  |
| --- | --- | --- | --- |
| Code | Value | Code | Value |
| 00000 | -90 | 010000 | -42 |
| 00001 | -87 | 010001 | -39 |
| 00010 | -84 | 010010 | -36 |
| 00011 | -81 | 010011 | -33 |
| 00100 | -78 | 010100 | -30 |
| 00101 | -75 | 010101 | -27 |
| 00110 | -72 | 010110 | -24 |
| 00111 | -69 | 010111 | -21 |
| 01000 | -66 | 011000 | -18 |
| 01001 | -63 | 011001 | -15 |
| 01010 | -60 | 011010 | -12 |
| 01011 | -57 | 011011 | -9 |
| 01100 | -54 | 011100 | -6 |
| 01101 | -51 | 011101 | -3 |
| 01110 | -48 | 011110 | 0 |
| 01111 | -45 | 011111 | reserved |

##### A.3.5.5.x2.11 ISM panning

NOTE: It is ffs whether for non-diegetic objects, for the IVAS a single parameter is used for the panning, reusing the azimuth coding [-90,90] deg.

ISM\_PANNING PI frame indicates the left and right panning gains (G\_l and G\_r, respectively) for the audio object. Figure A.1f22 shows the structure of ISM\_PANNING PI frame. Each panning gain is indicated with 4 bits which gives a resolution of , when the value for a panning gain has a range of [0, 1]. For more detailed information about ISM panning, see clause 5.6.4.4.

 0 1 2 3 4 5 6 7
 +-+-+-+-+-+-+-+-+
 | G\_l | G\_r |
 +-+-+-+-+-+-+-+-+

**Figure A.1f22: ISM\_PANNING PI data frame with left and right panning gains.**

[

##### A.3.5.5.x2.12 ISM proximity

ISM\_PROXIMITY PI frame indicates the distances between the audio objects in the scene. Figures A.1f23, A1.f34 and A.1f25 present ISM\_PROXIMITY PI data frames with two, three and four audio objects, respectively. In the PI frames, for example, p12 indicates the distance between the first and second audio object, p13 indicates the distance between the first and third audio object and so on. A single byte is reserved for each proximity component, which gives a resolution of approximately [0, 2.55] meters when each component indicates centimetres. Different resolutions can be achieved by reserving different number of bits per component.

 0 1 2 3 4 5 6 7
 +-+-+-+-+-+-+-+-+
 | p12 |
 +-+-+-+-+-+-+-+-+

**Figure A.1f23: ISM\_PROXIMITY PI data frame for two objects.**

 0 1 2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | p12 | p13 | p23 |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

**Figure A.1f24: ISM\_PROXIMITY PI data frame for three objects.**

 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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | p12 | p13 | p14 | p23 |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | p24 | p34 |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

**Figure A.1f25: ISM\_PROXIMITY PI data frame for four objects**

][

### A.3.5.5x Reverse direction PI data types

#### A.3.5.5x.x1 Orientation PI data (reverse direction)

##### A.3.5.5x.x1.x1 Orientation data structures

[The orientation data structures in the reverse direction consist of a 32-bit timestamp and an orientation data structure as defined in clause A.3.5x2.1.1.1. The 32-bit timestamp is in RTP timescale and should be based on the media timeline of the RTP clock of the playback device's received media. This allows a sender receiving this PI data to dejitter received orientations and estimate the latency at which a sender's orientations are applied.]

For the orientation data structure, see clause A.3.5x2.1.1.1.

##### A.3.5.5x.x1.x2 Playback device orientation

PLAYBACK\_DEVICE\_ORIENTATION PI data describes the orientation of the playback device with respect to the frontal direction of the device. The frontal direction refers to the playback device orientation at the beginning of playback, i.e. (w=0, x=1, y=0, z=0) in quaternions. Playback device orientation describes the deviation from the frontal direction. PLAYBACK\_DEVICE\_ORIENTATION PI type is used for feedback signalling.

The latest received PLAYBACK\_DEVICE\_ORIENTATION PI data is used until a new PLAYBACK\_DEVICE\_ORIENTATION PI data is received.

##### A.3.5.x.x1.x3 Head orientation

The HEAD\_ORIENTATION PI data describes the listener orientation and follows the description in clause 7.4.2.2 (Listener orientation), i.e., the head frontal direction is (w=0, x=1, y=0, z=0) in quaternions. HEAD\_ORIENTATION PI type is used for feedback signalling.

The latest received HEAD\_ORIENTATION PI data is used until a new HEAD\_ORIENTATION PI data is received.

[

NOTE: Split Rendering support in this payload format is under construction. The following parameters are candidates that allow signalling of a split rendering session that follows the agreed ISAR feature of the IVAS codec. Those parameters will not be supported in the initial CR to 26.114.

NOTE: Listener position could be transmitted as cartesian or spherical coordinates. In case both are included, there would be need for two PI types, e.g., LISTENER\_POSITION\_CARTESIAN and LISTENER\_POSITION\_SPHERICAL.

#### A.3.5.5x.x2 Listener position

LISTENER\_POSITION PI frame indicates the listener position in 3D space in cartesian coordinates as in figure A.1f15. Figure A.1f15 shows a general position PI data structure as cartesian coordinates (X, Y, Z). Each component is a 16 bitssigned integer in units of 0.01 metres, which gives a maximum distance of . This gives a range of approximately [-327.68, 327.68]] meters. The cartesian position coordinates follow the representation presented in clause 7.4 (Rendering control), where the x-axis points towards front, the y-axis points towards left and the z-axis points towards up.

 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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | X | Y |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Z |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

**Figure A.1f15: General position PI data frame with cartesian coordinates.**

]

#### A.3.5.5x.x3 Disable or enable device orientation compensation

Device orientations can be compensated in the transmitted audio by a sender, see DEVICE\_ORIENTATION\_COMPENSATED in clause A.3.5x2.1.1.3. In some cases, the receiver may want to disable the compensation done by the sender, i.e., to receive the uncompensated audio signal. This can be requested with a DISABLE\_DEVICE\_ORIENTATION\_COMPENSATION PI frame. The request indicates that device orientation compensations applied at the sender side should be disabled. The sender is not mandated to follow the request, but the request should be followed if possible. It is recommended that a request for enabling/disabling device orientation compensation by the receiver device results in a response from the sender device with the sender device transmitting the device orientation PI data. The receiver device can determine based on the received PI data if the request was implemented.

For example, if a sender is compensating device orientations in the transmitted audio (indicated by DEVICE\_ORIENTATION\_COMPENSATED PI frames), the receiver may request the sender to stop compensating device orientations in the transmitted audio by sending a DISABLE\_DEVICE\_ORIENTATION\_COMPENSATION PI data to the sender. The sender should then stop compensating the device orientations in the transmitted audio and may send the device orientations to the receiver with DEVICE\_ORIENTATION\_UNCOMPENSATED PI data. ENABLE\_DEVICE\_ORIENTATION\_COMPENSATION PI data can be used to request the opposite of the above, i.e., to request to enable applying device orientation compensation at a sender.

#### A.3.5.5x.x4 Reverse PI latency

A playback device sending reverse PI data may experience the result of its sent data by receiving the corresponding data in forward direction as forward PI data. This allows calculation of the time elapsed between the sent reverse PI data and received forward PI data. Such calculated PI latency is valuable for both the playback and the recording device to apply prediction when creating reverse PI data or when applying orientation data. This PI latency may be sent back the recording device using the PI\_LATENCY PI frames. Figure A.1f17 shows the data structure of a PI\_LATENCY frame, which consists of a signed 32-bit integer for the measured latency in RTP ticks, i.e. at a 16kHz clock rate. Positive numbers indicate an experienced latency, negative numbers can occur if prediction has been applied.

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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | latency |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure A.1f17: PI data frame with latency indication.

][

### A.3.5.5x2 Resilience against packet losses and redundant PI data for IVAS

NOTE: Describing resilience for packet loss for audio data is ffs. In 26.114 such discussion (at least for audio) could be present, including adding information on CMR resilience. Essentialty of PI data would dictate whether redundancy is needed. Error concealment for PI data is ffs.

IVAS PI data may in some cases be particularly sensitive to packet losses. An RTP sender [may] repeat the PI data in a few RTP packets to create resilience against packet losses.

Allowing redundant PI data will require adding additional bandwidth.

To reduce the impact of packet losses on media data for IVAS, an RTP sender may repeat the PI data in several RTP packets.

If several audio frames are included in the RTP packet and the PI data from a previous RTP packet and if the PI data needs to be synchronized with an audio frame of the previously transmitted RTP packet, then the RTP time stamp is calculated from the media time of the oldest PI data and one or more NO\_DATA frames is included before the first audio frame.

NOTE: This is needed so that the receiver can calculate the correct media time of the audio frame(s).

][

### A.3.5.5x3 Fragmentation of large payloads

To avoid IP fragmentation packet size should be below the transport channel's MTU size. Thus large payloads should be avoided. Some large payloads result from aggregation of multiple IVAS frames and PI data which should be split into several smaller payloads to stay below MTU size limitations at natural fragmentation points, for example on IVAS frame data and PI data borders.

NOTE: The details are FFS, for 768 kbps, there may be no natural fragmentation points.

]

# A.4 Payload Format Parameters

## A.4.1 IVAS Media Type Registration

The media type for the IVAS codec is to be allocated from the standards tree. This clause defines parameters of the IVAS payload format. This media type registration covers real-time transfer via RTP and non-real-time transfers via stored files. All media type parameters defined in this document shall be supported. [Additional parameters and their handling are ffs.]

Media type name: audio

Media subtype name: IVAS

Required parameters: none

Optional parameters:

The parameters defined below apply to RTP transfer only:

**ptime**: see [r1].

**maxptime**: see [r1].

**dtx/dtx-recv**: as defined in Annex A of [3]. Applicability of the dtx parameter to all operation modes is ffs.

**max-red**: see [r5].

**channels**: The number of audio channels shall not be present.

NOTE: The use of the channels parameter as defined in [r4] does not permit signaling all IVAS Immersive mode coded formats; formats need to be derived from the cf/cf-send/cf-recv parameters.

**ivas-mode-switch**: This parameter defines the mode at the start or update of the session for the send and the receive directions. Permissible values are 0 and 1. If ivas-mode-switch is 0 or not present, IVAS Immersive mode is used. If ivas-mode-switch is 1, depending on the setting of evs-mode-switch, EVS Primary or AMR-WB IO mode is used The mode initially used in the session may later be modified by the adaptation mechanisms present in this specification.

**cmr:** As defined in Annex A of [3] for the EVS Primary and AMRWB-IO modes. For IVAS Immersive modes the bit rate, bandwidth and format requests are disabled when cmr is -1. The bitrate, bandwidth and format requests are enabled when cmr is 0 or the cmr parameter is not present. When cmr is 1 the bit rate requests using the initial E byte shall be present in every packet (but may be NO\_REQ); format and bandwidth requests for IVAS Immersive modes are optional when cmr is 1.

The following parameters are applicable only to IVAS Immersive operation:

NOTE: IVAS computational complexity and memory demands of depend on the setting of the following parameters for source codec bit rate, audio bandwidth, and coded format; in addition, factors beyond the signaling, such as complexity of a specific implementation and the (rendered) output format may be significant. Further definition of complexity levels or guidelines is ffs.

**ibr**: Specifies the range of source codec bitrate for IVAS Immersive mode in the session, in kilobits per second, for the direction specified by the session directionality attribute or the suffix. The br parameter can either have: a single bitrate (ibr1); or a hyphen-separated pair of two bitrates (ibr1-ibr2). If a single value is included, this bitrate, ibr1, is used. If a hyphen-separated pair of two bitrates is included, ibr1 and ibr2 are used as the minimum bitrate and the maximum bitrate respectively. ibr1 shall be smaller than ibr2. ibr1 and ibr2 have a value from the set in Table 4.2-2 of the present document. If none of these parameters is present, all bitrates consistent with the IVAS codec capabilities are allowed in the session.

**ibr-send/ibr-recv**: ibr parameter in send or receive direction.

**ibw**: Specifies the audio bandwidth for IVAS Immersive modes to be used in the session, for the direction specified by the session directionality attribute or the suffix. ibw has a value from the set: wb, swb, fb, wb-swb, and wb-fb. wb, swb, and fb represent wideband, super-wideband, and fullband respectively, and wb-swb, and wb-fb represent all bandwidths from wideband to super-wideband, and fullband respectively. If none of these parameters is present, all bandwidths consistent with the negotiated bitrate(s) are allowed in the session.

**ibw-send/ibw-recv**: ibw parameter in send or receive direction.

**cf**: Specifies the IVAS Immersive mode coded-format (cf) transmitted in the IVAS Immersive mode frames in the session. IVAS coded format corresponds to the format represented in the IVAS Immersive mode coded frames, which is generally the input format to the encoder. The cf parameter is a list of supported comma-separated IVAS Immersive mode coded formats in the order of preference, using the identifiers from Table A.4h of the present document (column "Identifier"). Selection of the format is application-specific and out of scope of this document. EVS frames in the session are in mono format; switching to mono shall be possible.

[For SR format, the following applies: While the formats offered by the offererer may be a list containing SR and other formats, the answer shall either exclusively contain SR or a set of the other offered formats excluding SR. A combination of SR with other formats is not permissible.]

Table A.4h: IVAS coded-format

|  |  |  |
| --- | --- | --- |
| Identifier | Full Name | Clause |
| Stereo | Stereo Operation | 4.2.3 |
| SBA | Scene-based Audio (SBA, Ambisonics) Operation | 4.2.4 |
| MASA | Metadata-assisted Spatial Audio (MASA) Operation | 4.2.5 |
| ISM | Objects (Independent Streams with Metadata, ISM) Operation | 4.2.6 |
| MC | Multi-Channel (MC) Operation | 4.2.7 |
| OMASA | Combined Objects and MASA (OMASA) Operation | 4.2.9 |
| OSBA | Combined Objects and SBA (OSBA) Operation | 4.2.8 |
| [SR | Split rendering Operation | 7.6] |

Mono is not listed as an IVAS Immersive mode coded-format as EVS is always supported and shall be used for mono.

NOTE: IVAS is self-contained for all IVAS Immersive mode coded formats [except SR and mono] (which don't include mono); Mono is supported by using EVS.

[The exact syntax and identifiers may be modified and are ffs. This includes handling of binaural audio and sub-formats.]

**cf-send/cf-recv**: cf parameter in send or receive direction

**pi-types**: Specifies the supported PI data types for the session. The pi-types parameter is a list of supported comma-separated PI data types using the SDP indications listed in tables A.1x4, A.1x5 and A.1x6. If the pi-types parameter is not present, PI data is not enabled for the session.

**pi-types-send/pi-types-recv**: pi-types parameter in send or receive direction.

**pi-br**: Specifies the maximum bitrate for the PI data section (excluding the E-bytes for indication) for the session in kilobits per second. The parameter indicates the maximum bitrate for the PI data. If pi-br parameter is not present, a default value of [10] (kbps) shall be used.

**pi-br-send/pi-br-recv**: pi-br parameter in send or receive direction.

[

NOTE: Split Rendering support in this payload format is under construction. The following parameters are candidates that allow signalling of a split rendering session that follows the agreed ISAR feature of the IVAS codec. Those parameters will not be supported in the initial CR to 26.114.

**sr-dof**: Specifies the number of degrees of freedom supported in the head-tracked split rendering session. Permissive values are -1, 0, 1, 2, 3. [A value of -1 means that respective stream will be a non-diegetic stream in which the pre-renderer does not expect head-tracker data/does not take such data into account during pre-rendering.] A value in the range of 0 - 3 means that the pre-renderer expects head-tracker data, conveyed by PI frames or some other mechanism. A value of D > 0 means that metadata is generated and transmitted in the SR bitstream allowing the post-renderer to make pose corrections of the binaural audio in D degrees of freedom. D=1 means support of yaw pose corrections only, D=2 means support of yaw and pitch pose corrections, D=3 means support of pose corrections around all 3 axes. [The first value may be followed separated by a comma by a second value respective a second non-diegetic stream. The only permissible value for that stream is -1.]
If sr-dof is not present in a negotiated SR session, it defaults to 3.

**sr-nohq:** Specifies that high-quality split renderer metadata calculation is not used. If not present, high-quality calculation is used in a split rendering session with 3 DOF split rendering.

**sr-tc:** Specifies the codec format for the binaural transport channels. Permissible values are ‘LCLD’ and ‘LC3plus’. This parameter must be present in a negotiated split rendering session. When LC3plus is used in a session with split rendering, the fdi parameter shall also be present.

**fdi:** As specified in [X1] when LC3plus is used in a negotiated split rendering session. Shall be set to "1" or "2", depending on the split rendering configuration.

**bwr:** As specified in [X1] when LC3plus is used in a negotiated split rendering session. Shall be set to "fb" for split rendering sessions.

NOTE: A single split rendering SDP parameter bundling everything is ffs.

]

[

**pmode**: Specifies the packetization mode that restricts the properties of an RTP payload type to fit to the capabilities of an implementation. The value of pmode shall be 1. If this parameter is not present, it will be assumed to be 1.

NOTE: Subsequent versions may add further packetization modes that enlarge the payload format functionality.

]

NOTE: Additional RTP payload elements to e.g. support fine-granular format requests, fragmentation or multi-stream handling are ffs.

The following parameters are applicable only to EVS Primary and AMR-WB IO modes:

**evs-mode-switch**: as defined in Annex A of [3]. If ivas-mode-switch is 0 or not present, evs-mode-switch should not be present and shall be ignored.

**hf-only**: as specified in Annex A of [3] except that the default and only allowed value of hf-only shall be 1 in this payload format. As the only allowed value for this parameter is 1 it is not required to include this parameter.

NOTE: There is no compact format support in this payload format, contrary to the EVS payload format in Annex A of [3] that enables the compact format by default.

**ch-send:** Shall not be present. The EVS modes in this payload format shall be mono-only

**ch-recv:** Shall not be present. The EVS modes in this payload format shall be mono-only.

The following parameters are applicable only to EVS Primary modes:

**br**: as defined in Annex A of [3]

**br-send**: as defined in Annex A of [3]

**br-recv**: as defined in Annex A of [3]

**bw**: as defined in Annex A of [3]

NOTE: Narrow-band is not supported for IVAS operation

**bw-send**: as defined in Annex A of [3]

**bw-recv**: as defined in Annex A of [3]

**ch-aw-recv**: as defined in Annex A of [3]

The following parameters are applicable only to EVS AMR-WB IO modes:

**mode-set**: as defined in Annex A of [3]

**mode-change-period**: see [r5].

**mode-change-capability**: as defined in Annex A of [3]

**mode-change-neighbor**: see [r5].

CHANGE 2

## A.4.2 Mapping media type parameters into SDP

The information carried in the media type specification has a specific mapping to fields in the Session Description Protocol (SDP) [r1], which is commonly used to describe RTP sessions. When SDP is used to specify sessions employing the IVAS codec, the mapping is as follows:

- The media type ("audio") goes in SDP "m=" as the media name.

- The media subtype (payload format name) goes in SDP "a=rtpmap" as the encoding name. The RTP clock rate in "a=rtpmap" shall be 16000, and the encoding parameters (number of channels) shall be omitted.

- The parameters "ptime" and "maxptime" go in the SDP "a=ptime" and "a=maxptime" attributes, respectively.

- Any remaining parameters go in the SDP "a=fmtp" attribute by copying them directly from the media type parameter string as a semicolon-separated list of parameter=value pairs.

Mapping to fields in SDP is specified in clause 6 of [r2].

CHANGE 3

## A.4.3 Detailed Description of Usage of SDP Parameters

### A.4.3.1 Offer-Answer Model Considerations

The following considerations apply when using SDP Offer-Answer procedures to negotiate the use of IVAS payload in RTP:

NOTE: Split Rendering support in this payload format is under construction. Specific SR parameter would be added to the offer-answer model considerations once they stabilized.

**hf-only**: Shall not be included in the SDP offer. The answerer shall include this parameter only if it is set to 1 in the SDP offer. If the value in the SDP offer is not equal to 1, the payload type shall be rejected.

**ivas-mode-switch**: When ivas-mode-switch is not offered for a payload type, the answerer may include ivas-mode-switch for the payload type in the SDP answer. When ivas-mode-switch is offered for a payload type and the payload type is accepted, the answerer shall not modify or remove ivas-mode-switch for the payload type in the SDP answer.

**cmr**: When cmr is not offered for a payload type, the answerer may include cmr for the payload type in the SDP answer. When cmr is offered for a payload type and the payload type is accepted, the answerer shall not modify or remove cmr for the payload type in the SDP answer.

**ibr**: When the same bitrate or bitrate range is defined for the send and the receive directions, ibr should be used but ibr-send and ibr-recv may also be used. ibr can be used even if the session is negotiated to be sendonly, recvonly, or inactive. For sendonly session, ibr and ibr-send can be interchangeably used. For recvonly session, ibr and ibr-recv can be interchangeably used. When ibr is not offered for a payload type, the answerer may include ibr for the payload type in the SDP answer. When ibr is offered for a payload type and the payload type is accepted, the answerer shall include ibr in the SDP answer which shall be identical to or a subset of ibr for the payload type in the SDP offer.

**ibr-send**: When ibr-send is not offered for a payload type, the answerer may include ibr-recv for the payload type in the SDP answer. When ibr-send is offered for a payload type and the payload type is accepted, the answerer shall include ibr-recv in the SDP answer, and the ibr-recv shall be identical to or a subset of ibr-send for the payload type in the SDP offer.

**ibr-recv**: When ibr-recv is not offered for a payload type, the answerer may include ibr-send for the payload type in the SDP answer. When ibr-recv is offered for a payload type and the payload type is accepted, the answerer shall include ibr-send in the SDP answer, and the ibr-send shall be identical to or a subset of ibr-recv for the payload type in the SDP offer.

**ibw**: When the same bandwidth or bandwidth range is defined for the send and the receive directions, ibw should be used but ibw-send and ibw-recv may also be used. ibw can be used even if the session is negotiated to be sendonly, recvonly, or inactive. For sendonly session, ibw and ibw-send can be interchangeably used. For recvonly session, ibw and ibw-recv can be interchangeably used. When ibw is not offered for a payload type, the answerer may include ibw for the payload type in the SDP answer. When ibw is offered for a payload type and the payload type is accepted, the answerer shall include ibw in the SDP answer, which shall be identical to or a subset of ibw for the payload type in the SDP offer.

**ibw-send**: When ibw-send is not offered for a payload type, the answerer may include ibw-recv for the payload type in the SDP answer. When ibw-send is offered for a payload type and the payload is accepted, the answerer shall include ibw-recv in the SDP answer, and the ibw-recv shall be identical to or a subset of ibw-send for the payload type in the SDP offer.

**ibw-recv** When ibw-recv is not offered for a payload type, the answerer may include ibw-send for the payload type in the SDP answer. When ibw-recv is offered for a payload type and the payload is accepted, the answerer shall include ibw-send in the SDP answer, and the ibw-send shall be identical to or a subset of ibw-recv for the payload type in the SDP offer.

**cf**: The SDP offer [shall] list at least one but may list several IVAS Immersive mode coded formats. The SDP answer shall include at least one IVAS Immersive mode coded format and should respond with the one most preferred coded format from the list in the SDP offer. If more than one format is present in the SDP answer, the first format shall be used at the start of a session and may only be modified by the adaptation mechanisms present in this specification. When the same IVAS Immersive mode coded formats are defined for the send and the receive directions, cf should be used but cf-send and cf-recv may also be used. For sendonly session, cf and cf-send can be interchangeably used. For recvonly session, cf and cf-recv can be interchangeably used.

**cf-send**: When cf-send is offered for a payload type and the payload type is accepted, the answerer shall include cf-recv in the SDP answer, and the cf-recv shall be identical to or a subset of the cf-send parameter for the payload type in the SDP offer.

**cf-recv** When cf-recv is offered for a payload type and the payload type is accepted, the answerer shall include cf-send in the SDP answer, and the cf-send shall be identical to or a subset of the cf-recv parameter for the payload type in the SDP offer.

**pi-types**: The SDP offer shall list at least one but may list several supported pi types when pi data is enabled in the offer. When one or more of the offered pi types are supported, the SDP answer shall be identical to or a subset of the pi types listed in the SDP offer. When the same pi types are defined for the send and the receive directions, pi-types should be used but pi-types-send and pi-types-recv may also be used. For sendonly session, pi-types and pi-types-send can be interchangeably used. For recvonly session, pi-types and pi-types-recv can be interchangeably used. When none of the offered pi-types is supported, the answerer shall not include pi-types in the SDP answer.

**pi-types-send:** When pi-types-send is offered in the SDP offer and it is accepted, the answerer shall include pi-types-recv in the SDP answer, and the pi-types-recv shall be identical to or a subset of the pi-types-send parameter in the SDP offer.

**pi-types-recv**: When pi-types-recv is offered in the SDP offer and it is accepted, the answerer shall include pi-types-send in the SDP answer, and the pi-types-send shall be identical to or a subset of the pi-types-recv parameter in the SDP offer.

**pi-br**: When the same bitrate is defined for the send and the receive directions, pi-br should be used but pi-br-send and pi-br-recv may also be used. pi-br can be used even if the session is negotiated to be sendonly, recvonly, or inactive. For sendonly session, pi-br and pi-br-send can be interchangeably used. For recvonly session, pi-br and pi-br-recv can be interchangeably used. When pi-br is not offered in the SDP offer, the answerer shall not include pi-br in the SDP answer. When pi-br is offered in the SDP offer and it is accepted, the answerer shall include pi-br in the SDP answer which shall be identical or lower than pi-br in the SDP offer.

**pi-br-send**: When pi-br-send is offered in the SDP offer and it is accepted, the answerer shall include pi-br-recv in the SDP answer, and the pi-br-recv shall be identical or lower than pi-br-send in the SDP offer.

**pi-br-recv**: When pi-br-recv is offered in the SDP offer and it is accepted, the answerer shall include pi-br-send in the SDP answer, and the pi-br-send shall be identical or lower than pi-br-recv in the SDP offer.

The offer-answer considerations for the remaining EVS parameters are as described in TS 26.445 Annex A.3.3.1 [3].

]

END OF CHANGES