**3GPP TSG- Meeting #**

**, , -**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.3* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  |  | **CR** |  | **rev** |  | **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:*** | S4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***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 can be 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:*** | | Split rendering support is missing for the IVAS payload format. Many features enabled with new PI types. Readability for certain figures and payload parsing needs improvement. E byte would be full with the current types without extension capabilites. Negotiation parameter ims is colliding with similarly named parameter and needs renaming. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | Added Split rendering support, support for reverse direction PI data, readability improvements for figures, payload parsing clause, E byte ET field extended to 3 bits, many new PI data types, ims/ivas-mode-switch parameter renamed to mono-init. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | There would be no Split rendering support in IVAS payload, subsequent E byte signalling would be full without future extensibility, similarly named parameters would be confusing. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | A.3, A.4 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | | **x** |  | Other core specifications | | | | TS 26.114 | | |
| ***affected:*** | |  | **x** | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  | **x** | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

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

- IVAS split rendering

- 48 KHz sampling rate

- Bitrates ranging from 256 kbps to 512 kbps

- 20 ms, 10 ms and 5 ms frame durations

- multiple frames per RTP payload

- 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 [34]. This IVAS RTP payload format uses the fields of the RTP header in a manner consistent with the usages in [34].

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 [36] and EVS codecs [3]; having a unique clock rate across all payload types of one media avoids the issues described in [37].

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 duration of one IVAS frame in split rendering mode can be 20, 10 or 5 ms and hence the media time is increased for each successive IVAS split rendering frame of an RTP packet by 320, 160 or 80 ticks respectively. 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.3.5.4.

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 [35].

## 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;

In Split rendering mode, IVAS decoder/renderer generates frames representing 20, 10 or 5 ms of speech or audio data (frame length is decided based on the SDP parameter **sr-tc-fr** negotiation)

### 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

### 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.3.3.3.2-1.

|  |
| --- |
| 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.3.3.3.2-1.

Table A.3.3.3.2-1: 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 | IVAS-SR |
| 0 | 1 | 1111 | IVAS 5.2 kbps SID |

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.

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 set to 0 and the Unused/IVAS indicator bit set to 1. In this 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|D|C|SR\_BR| res | +-+-+-+--+--+-+-+-+ |

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

D (1 bit): Identifier information identifying the split rendering stream associated with the ToC as diegetic (D=1) or non-diegetic (D=0).

C (1 bit): Split rendering transport codec bit as defined in Table A.3.3.3.2-4.

SR-BR (2 bits): Bit rate index as defined in Table A.3.3.3.2-3.

**Table** **A.3.3.3.2-3: Indicated IVAS split rendering bit rate (SR-BR)**

|  |  |
| --- | --- |
| **SR-BR** | **Indicated IVAS SR bit rate** |
| 00 | Reserved |
| 01 | IVAS SR 256 kbps |
| 10 | IVAS SR 384 kbps |
| 11 | IVAS SR 512 kbps |

Table A.3.3.3.2-4: C field in SR-ToC byte.

|  |  |
| --- | --- |
| C | Definition |
| 0 | LCLD |
| 1 | LC3plus |

NOTE: When split rendering transport codec is changed as per C bit, the codec parameters like frame size indicator, fdi and bwr are used as per SDP negotiation.

Any reserved bits in the ToC byte(s) shall be set to zero by the media sender and ignored by the media receiver.

#### 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.3.3.3.3.1-1. E bytes contain extra information. Specific rules on E byte and ToC byte placement in the payload and parsing of the payload header are specified in clause A.3.3.3.4.

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

Figure A.3.3.3.3.1-2: Void.

##### 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.3.3.3.2-1.

Table A.3.3.3.2-1: 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 | Reserved |
| 1111 | NO\_REQ |
|  | | |

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.

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.3.3.3.2-1.

NOTE: When operating in IVAS Immersive mode, a received EVS CMR (T=000..110) is be interpreted as a request to switch to EVS operation mode. When operating in EVS mode, a received IVAS (Immersive) CMR (T=111) is be interpreted as a request to switch to IVAS Immersive operation mode.

The codec mode request indicated in the 4-bit-CMR shall comply with the media type parameters (the allowed bit-rates for IVAS) that are negotiated for the session. When a 4-bit-CMR is received, requesting a bit-rate that does not comply with the negotiated media parameters, it shall be ignored.

##### 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, coded format, or to indicate the presence of PI data or Split renderer request in the payload as described in the following clauses. 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 (3 bits): Type of subsequent E byte as indicated in Table A.3.3.3.3.3-1.

Table A.3.3.3.3.3-1: ET field in a subsequent E byte

|  |  |
| --- | --- |
| ET | Definition |
| 000 | Bandwidth Request |
| 001 | Format Request |
| 010 | PI indication |
| 011 | Split Renderer Request |
| 100 to 111 | Reserved for future use |

All reserved bits in any subsequent E-byte shall be set to zero by the media sender and ignored by the media receiver. Similarly, any value denoted ‘reserved’ shall be ignored.

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  +-+-+-+-+-+-+-+-+ |1|0 0 0|res|BW | +-+-+-+-+-+-+-+-+ |

Figure A.3.3.3.3.3.2-1: Subsequent E byte structure for bandwidth request (ET=000)

The contents of this subsequent E byte has following meaning:

BW (2 bits): Requested bandwidth as indicated in Table A.3.3.3.3.3.2-1.

Table A.3.3.3.3.3.2-1: BW field in a subsequent E byte

|  |  |
| --- | --- |
| BW | Definition |
| 00 | WB |
| 01 | SWB |
| 10 | FB |
| 11 | NO\_REQ |

The bandwidth request indicated in the BW field shall comply with the media type parameters (allowed bandwidths for IVAS) that are negotiated for the session. When a BW field is received, requesting a bandwidth that does not comply with the negotiated media parameters, it shall be ignored.

A.3.3.3.3.3.3 Coded Format Request

An E byte with ET=001 (Format Request) shall use coded format requests defined as in Figure A.3.3.3.3.3.3-1.

|  |
| --- |
| 0 1 2 3 4 5 6 7  +-+-+-+-+-+-+-+-+ |1|0 0 1|S| FMT | +-+-+-+-+-+-+-+-+ |

Figure A.3.3.3.3.3.3-1: Subsequent E byte structure for coded format request (ET=001)

The contents of this subsequent E byte has following meaning:

S (1 bit): Subformat indication.

If S=0, the coded format request is fully defined in the current E-byte.

If S=1, the FMT field shall be set to ‘111’ and a receiver shall ignore these bits.

Specifically, when S = 0, the FMT bits have the following meaning:

FMT (3 bits): Requested coded format as indicated in Table A.3.3.3.3.3.3-1.

Table A.3.3.3.3.3.3-1: FMT field in a subsequent E byte (when S=0)

|  |  |
| --- | --- |
| FMT | Definition |
| 000 | Stereo |
| 001 | SBA |
| 010 | MASA |
| 011 | ISM |
| 100 | MC |
| 101 | OMASA |
| 110 | OSBA |
| 111 | NO\_REQ |

NOTE: Mono is not included in Table A.3.3.3.3.3.3-1 as mono coding in IVAS is handled by the EVS modes.

When S = 1, the FMT bits have no meaning and an extra byte shall be inserted immediately after the current E-byte to request a subformat (see Figure A.3.3.3.3.3.3-2).

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

Figure A.3.3.3.3.3.3-2: Extra byte structure to indicate a subformat request immediately after a coded format request (ET=001)

The contents of this extra byte has following meaning:

subFMT (6 bits): Requested coded subformat as indicated in Table A.3.3.3.3.3.3-2.

res (2 bits): Reserved bits.

Table A.3.3.3.3.3.3-2: subFMT field in the extra byte when S=1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| subFMT | Definition |  | subFMT | Definition |
| 000000 | FOA planar | 100000 | OMASA ISM1 1TC |
| 000001 | HOA2 planar | 100001 | OMASA ISM2 1TC |
| 000010 | HOA3 planar | 100010 | OMASA ISM3 1TC |
| 000011 | FOA | 100011 | OMASA ISM4 1TC |
| 000100 | HOA2 | 100100 | OMASA ISM1 2TC |
| 000101 | HOA3 | 100101 | OMASA ISM2 2TC |
| 000110 | MASA1 | 100110 | OMASA ISM3 2TC |
| 000111 | MASA2 | 100111 | OMASA ISM4 2TC |
| 001000 | ISM1 | 101000 | OSBA ISM1 FOA planar |
| 001001 | ISM2 | 101001 | OSBA ISM2 FOA planar |
| 001010 | ISM3 | 101010 | OSBA ISM3 FOA planar |
| 001011 | ISM4 | 101011 | OSBA ISM4 FOA planar |
| 001100 | ISM1 extended metadata | 101100 | OSBA ISM1 FOA |
| 001101 | ISM2 extended metadata | 101101 | OSBA ISM2 FOA |
| 001110 | ISM3 extended metadata | 101110 | OSBA ISM3 FOA |
| 001111 | ISM4 extended metadata | 101111 | OSBA ISM4 FOA |
| 010000 | MC 5.1 | 110000 | OSBA ISM1 HOA2 planar |
| 010001 | MC 7.1 | 110001 | OSBA ISM2 HOA2 planar |
| 010010 | MC 5.1.2 | 110010 | OSBA ISM3 HOA2 planar |
| 010011 | MC 5.1.4 | 110011 | OSBA ISM4 HOA2 planar |
| 010100 | MC 7.1.4 | 110100 | OSBA ISM1 HOA2 |
| 010101 | Reserved | 110101 | OSBA ISM2 HOA2 |
| 010110 | Reserved | 110110 | OSBA ISM3 HOA2 |
| 010111 | Reserved | 110111 | OSBA ISM4 HOA2 |
| 011000 | Reserved | 111000 | OSBA ISM1 HOA3 planar |
| 011001 | Reserved | 111001 | OSBA ISM2 HOA3 planar |
| 011010 | Reserved | 111010 | OSBA ISM3 HOA3 planar |
| 011011 | Reserved | 111011 | OSBA ISM4 HOA3 planar |
| 011100 | Reserved | 111100 | OSBA ISM1 HOA3 |
| 011101 | Reserved | 111101 | OSBA ISM2 HOA3 |
| 011110 | Reserved |  | 111110 | OSBA ISM3 HOA3 |
| 011111 | Reserved |  | 111111 | OSBA ISM4 HOA3 |

The coded format request indicated in the FMT or subFMT field shall comply with the media type parameters (allowed coded formats for IVAS) that are negotiated for the session. When an FMT or subFMT field is received, requesting a coded format that does not comply with the negotiated media parameters, it shall be ignored.

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.3.3.3.3.3.4-1.

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

Figure A.3.3.3.3.3.4-1: Subsequent E byte structure to indicate PI data presence in the payload (ET=010)

A.3.3.3.3.3.5 Split Renderer Configuration Request

Split Renderer configuration requests are defined as shown in Figure A.3.3.3.3.3.5-1. This subsequent E byte is used to request diegetic or non-diegetic support. Diegetic bit D=0 corresponds to requesting a non-headtrackable stereo, 0-DoF binaural, or non-diegetic stream while Diegetic bit D=1 corresponds to requesting a headtrackable diegetic stream with pose correction possibility at the receiver. In addition, bits Y (yaw), P (pitch), R (roll) are used to request transmission of pose correction metadata for correction around the corresponding axes or to request disabling transmission of such pose correction metadata around the corresponding axes.

The following control fields are defined with the following meaning:

D (1 bit): If the stream is negotiated as a diegetic split rendering stream, the D bit can be used to control if the stream should have diegetic support. D=’1’ requests enabling diegetic support; D=’0’ requests disabling it.

Y (1 bit): Request transmission of pose correction metadata for correction around yaw axis (z-axis). ‘1’ enables pose correction around this axis, ‘0’ disables it.

P (1 bit): Request transmission of pose correction metadata for correction around pitch axis (y-axis). ‘1’ enables pose correction around this axis, ‘0’ disables it.

R (1 bit): Request transmission of pose correction metadata for correction around roll axis (x-axis). ‘1’ enables pose correction around this axis, ‘0’ disables it.

The following rules apply:

* The number of axis around which pose correction is supported may be limited during session setup. A split renderer configuration request shall not exceed this limit.
* When D bit is set to 0 then Y, P and R bit fields are treated as reserved fields and shall be set to 0.

|  |
| --- |
| 0 1 2 3 4 5 6 7  +-+-+-+-+-+-+-+-+ |1|0 1 1|D|Y|P|R| +-+-+-+-+-+-+-+-+ |

Figure A.3.3.3.3.3.5-1: Subsequent E byte structure for Split Renderer configuration request (ET=011)

A.3.3.3.3.3.6 Reserved for future use

Subsequent E byte with ET bits in the range from 100 – 111 are reserved for future use. The structure of such subsequent E bytes is as shown in Figure A.3.3.3.3.3.6-1.

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

Figure A.3.3.3.3.3.6-1: Subsequent E byte structure for (ET=100 - 111)

In this case, the bits 4-7 are reserved for future use.

E bytes reserved for future use shall be the last subsequent E bytes prior to ToC bytes. Legacy receiver implementations (not supporting the future use) shall treat the bytes following the first encountered future use E byte until a first ToC byte as follows: The H bit position shall be parsed and all other bit positions shall be ignored. The first byte with H bit set to 0 shall be treated as first ToC byte.

#### A.3.3.3.4 Placement of E bytes and ToC bytes and parsing of the payload header

If there is any E byte present, the first E byte shall be the initial E byte with the CMR which shall precede all subsequent E bytes and ToC bytes. There shall be at most one initial E byte in the payload. If an E byte relates to a given frame, then it shall directly precede the ToC byte that relates to that frame. E bytes not related to specific frames shall always appear only once and precede all ToC bytes.

Parsing of one payload header byte follows the state machine of Figure A.3.3.3.4-1.

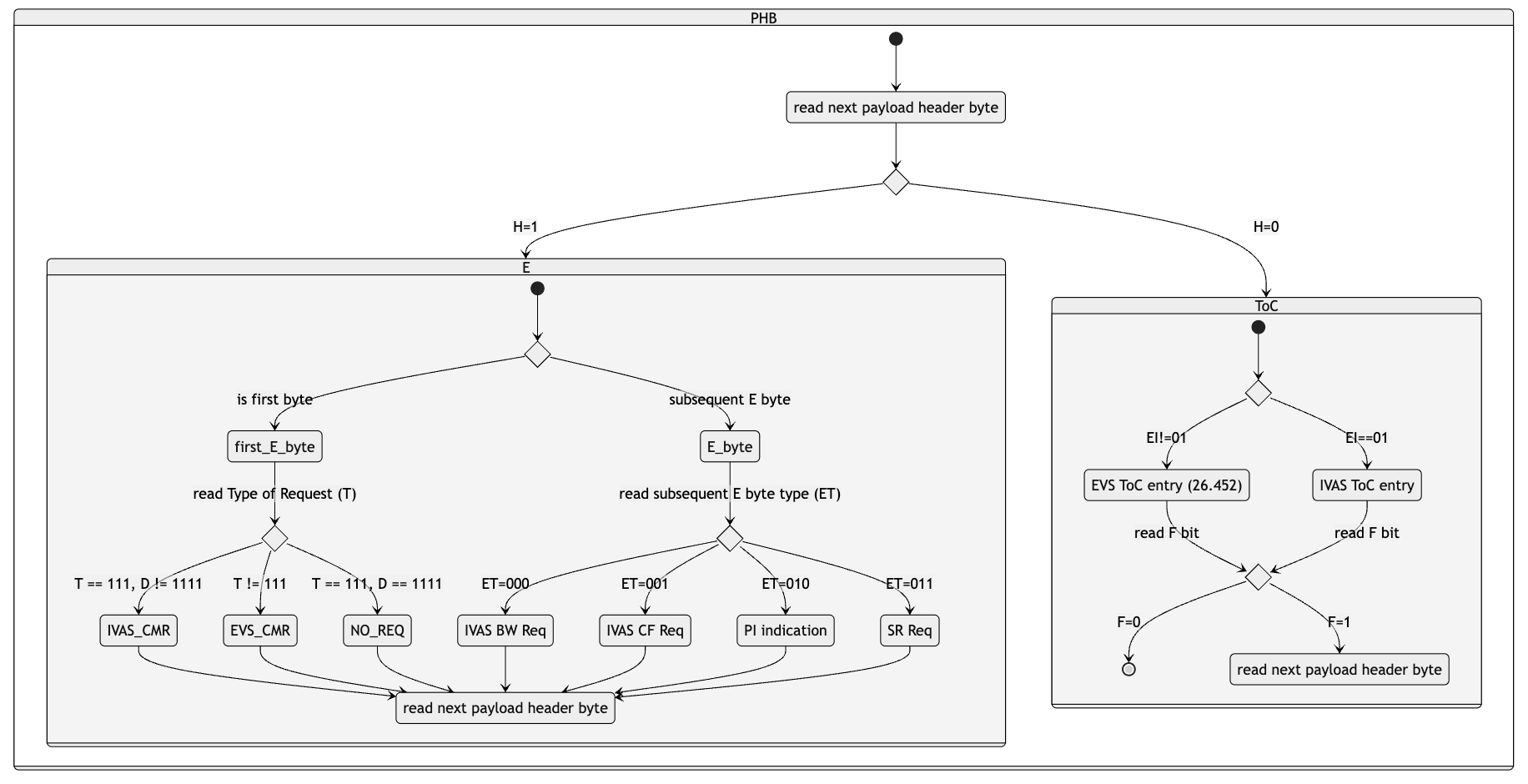


Figure A.3.3.3.4-1: State Machine for parsing a Payload Header Byte.

Figure A.3.3.3.4-2 presents a general payload structure for IVAS with two speech frames and multiple E bytes in the payload. Parsing of the payload header bytes follow the state machine of Figure A.3.3.3.4-1. The ToC bytes ToC1 (F bit set to 1) and ToC2 (F bit set to 0) identify the IVAS data frames “IVAS frame 1” and “IVAS frame 2” in the payload, respectively. The “Initial E byte” shall be a CMR byte and it shall be the first byte in the payload. The “Subsequent E byte 1” and “Subsequent E byte 2” may be of any subsequent E byte types listed in Table A.3.3.3.3.3-1. The “Subsequent E byte 3” shall be of any subsequent E byte type identified as frame specific in Table A.3.3.3.4-1 and the E byte relates to “IVAS frame 2”.

|  |
| --- |
| 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 2 3 4 5 6 7 8 9  H H H H F H  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |1| T | D |1| ET1 |x x x x|1| ET2 |x x x x|0|1|0 1| BR |1| ET3 |x x x x|…  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  \--------------/\--------------/\--------------/\--------------/\--------------/  Initial E byte Subsqnt E byte1 Subsqnt E byte2 ToC1 Subsqnt E byte3  H F  +-+-+-+-+-+-+-+-+-------------------------- ---+-------------------- ---+ …|0|0|0 1| BR | IVAS frame 1 ... | IVAS frame 2 ... |  +-+-+-+-+-+-+-+-+-------------------------- ---+-------------------- ---+  \--------------/  ToC2 |

Figure A.3.3.3.4-2: Payload structure with IVAS frames and E bytes in the payload.

Table A.3.3.3.4-1: Frame specific relations of subsequent E byte types.

|  |  |
| --- | --- |
| Subsequent E byte type | Frame specific |
| Bandwidth Request | No |
| Format Request | No |
| PI indication | No |
| Split Renderer Request | No |

NOTE: The payload structure in Figure A.3.3.3.4-2 is an example and the number and placement of the E bytes may vary. None of the currently specified subsequent E bytes can be frame specific.

## 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

The support of PI data is disabled by default and needs to be explicitly enabled using the pi-types parameter as defined in clause A.4.1.

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.

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.3.3.2-1.

|  |
| --- |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  | 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.3.5.2-1.

|  |
| --- |
| 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.3.5.2-1: PI data header.

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. See Table A.3.5.2-1.

PM (2 bits): PI frame marker bits indicate the associated audio frame for the PI data frame, see Table A.3.5.2-2.

PI type (5 bits): PI type bits indicate the type for the PI data as indicated in tables A.3.5.5-1 and A.3.5.5-2.

PI size (8 bits): PI size bits indicate the size for the PI data frame in bytes, see table A.3.5.2-3. A size of zero indicates that there is no PI data frame associated for the PI header indication.

Table A.3.5.2-1: PF bits for a PI header.

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

Table A.3.5.2-2: 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.3.5.2-2.



Figure A.3.5.2-2: State machine for parsing a PI header byte.

Table A.3.5.2-3 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.3.5.2-3: 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.

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.3.5.4 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.3.5.4 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.

### A.3.5.4 PI data handling during DTX

No audio frames are transmitted when DTX operation mode is determined by the media sender. Consequently, there is no RTP time stamp available to be associated with the transmitted PI data. If the PI data is obtained during DTX operation and needs to be transmitted as soon as possible, the PI data can be associated with the next transmitted SID frame or with a NO\_DATA frame. 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 timestamp of the transmitted packet is calculated from the media time of the oldest PI data included in the packet with one or more NO\_DATA frames included before the first audio frame.

If the transmission of the PI data is delayed, there can be multiple PI data frames of the same type (e.g., of device orientation type) waiting to be transmitted at the end of DTX operation. In this case, the latest or all PI data can be selected to be transmitted (e.g., the latest device orientation).

### A.3.5.5 Supported PI data types

Supported PI types are listed in tables A.3.5.5-1 and A.3.5.5-2 and described in the following subsections. Table A.3.5.5-1 lists PI types for forward direction signalling. Table A.3.5.5-1A lists the PI types for reverse direction signalling. Table A.3.5.5-2 lists additional PI types.

Table A.3.5.5-1: 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 in unit quaternions. | fsco | 8 | A.3.5.6.1.2 |
| 00001 | DEVICE\_ORIENTATION\_COMPENSATED | Describes the orientation of a device in unit quaternions. The orientation is compensated in the transmitted audio. | fdoc | 8 | A.3.5.6.1.3 |
| 00010 | DEVICE\_ORIENTATION\_UNCOMPENSATED | Describes the orientation of a device in unit quaternions. The orientation is not compensated in the transmitted audio. | fdou | 8 | A.3.5.6.1.3 |
| 00011 | ACOUSTIC\_ENVIRONMENT | Selects and optionally describes the acoustic environment. | face | 1,5 or 8 | A.3.5.6.2 |
| 00100 | AUDIO\_DESCRIPTION | Describes the content in the transmitted audio | faud | 1 to 5 | A.3.5.6.3 |
| 00101 | ISM\_NUM | Indicates the number of ISM(s). | finm | 1 | A.3.5.6.4.2 |
| 00110 | ISM\_ID | Indicates ID of each transported ISM. | fiid | Number of ISMs  x 1 | A.3.5.6.4.3 |
| 00111 | ISM\_GAIN | Describes gain factor for each ISM. | figa | Number of ISMs  x 1 | A.3.5.6.4.4 |
| 01000 | ISM\_ORIENTATION | Describes an orientation for each ISM. | fiso | Number of ISMs  x 8 | A.3.5.6.4.5 |
| 01001 | ISM\_POSITION | Describes a position of each ISM. | fipo | Number of ISMs  x 6 | A.3.5.6.4.6 |
| 01010 | ISM\_DISTANCE\_ATTENUATION | Describes distance attenuation for all ISMs. | fida | 3 or Number of ISMs  x 3 | A.3.5.6.4.7 |
| 01011 | ISM\_DIRECTIVITY | Describes directivity for each ISM. | fidr | 2 or  Number of ISMs  x 2 | A.3.5.6.4.8 |
| 01100 | DIEGETIC\_TYPE | Indicates if the audio is diegetic or non-diegetic. | fdit | 1 | A.3.5.6.5 |

**Table A.3.5.5-1A: 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 in Quaternions. | rpdo | 8 | A.3.5.7.1 |
| 10001 | HEAD\_ORIENTATION | Feedback. Describes the head orientation of the listener in Quaternions. | rhor | 8 | A.3.5.7.2 |
| 10010 | LISTENER\_POSITION | Feedback. Describes the position of the listener in 3D space. | rlip | 6 | A.3.5.7.3 |
| 10011 | DYNAMIC\_AUDIO\_SUPPRESSION | Describes receiver’s preference with respect to audio suppression | rdas | 2 | A.3.5.7.4 |
| 10100 | AUDIO\_FOCUS\_DIRECTION | Describes a direction of interest for the listener in Quaternions. | rafd | 8 | A.3.5.7.5 |
| 10101 | PI\_LATENCY | Round-trip latency for PI frames | rlat | 4 | A.3.5.7.6 |
| 10110 | R\_ISM\_ID | Indicates ID of each editing request ISM. | riid | Number of ISMs  x 1 | A.3.5.7.7.2 |
| 10111 | R\_ISM\_GAIN | An editing request for gain factor for each ISM. | riga | Number of ISMs  x 1 | A.3.5.7.7.3 |
| 11000 | R\_ISM\_ORIENTATION | An editing request for orientation for each ISM. | riso | Number of ISMs  x 8 | A.3.5.7.7.4 |
| 11001 | R\_ISM\_POSITION | An editing request for position for each ISM. | ripo | Number of ISMs  x 6 | A.3.5.7.7.5 |
| 11010 | R\_ISM\_DIRECTION | An editing request for direction for each ISM. | rido | Number of ISMs  x 2 | A.3.5.7.7.6 |

Table A.3.5.5-2: Additional PI types in an IVAS session.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type bits** | **PI type** | **Description** | **SDP indication** | **Size (bytes)** |
| 01101, 01110, 01111, 11011-11110 | 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.3.5.2-2. 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.3.5.6 Forward direction PI data types

#### A.3.5.6.1 Orientation PI data (forward direction)

##### A.3.5.6.1.1 Orientation data structures

Figure A.3.5.6.1.1-1 below shows PI orientation data structures in quaternions with 16 bits reserved for each component. The quaternion component values range from -1 to 1 according to Q15, in which 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.

NOTE: An orientation in Euler convention can be converted to quaternions before transmission, see clause 7.4.3.2 for the conversion operation and clause 7.4.3.1 for Euler angle definitions in 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   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  | W | X |  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  | Y | Z |  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |

**Figure A.3.5.6.1.1-1: PI orientation data as quaternions.**

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

##### A.3.5.6.1.2 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.3.5.6.1.3 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.3.5.6.2 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. The detailed description of room acoustics parameters is provided in clause 7.4.8.

The 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 simplified early reflections parameters include:

- 3D rectangular virtual room dimensions,

- Broadband energy absorption coefficient per wall.

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 containing only late reverb parameters (40 bits),

- a compact representation of the acoustic environment containing late reverb and simplified early reflections coefficients (64 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.3.5.6.2-1. Such an acoustic environment should be available upfront. It can be also provided using an AE PI data frame containing a 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.3.5.6.2-1: 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.

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 can be provided with or without early reflections coefficients. An AE data frame without early reflections is presented in Figure A.3.5.6.2-2.

|  |
| --- |
| 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.3.5.6.2-2: Acoustic environment PI data frame (ACOUSTIC\_ENVIRONMENT) containing a compact AE representation with late reverb parameters.

An AE data frame with early reflections is presented in Figure A.3.5.6.2-3.

|  |
| --- |
| 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 | dim x | dim y | dim z |ab0|ab1|ab2|ab3|ab4|ab5|  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |

Figure A.3.5.6.2-3: Acoustic environment PI data frame (ACOUSTIC\_ENVIRONMENT) containing a compact AE representation with late reverb and early reflection parameters.

The data frame including early reflections consists of an acoustic environment ID field (7 bits), three RT60 fields (5 bits each), three DSR fields (6 bits each), three room dimension (dim) fields (4 bits each) and six absorption coefficient (ab) fields summing up to 64 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.3.5.6.2-2 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.3.5.6.2-1.

Table A.3.5.6.2-1: 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 as shown in Table A.3.5.6.2-2.

Table A.3.5.6.2-2: 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.

The three room dimensions: length (x), width (y), and height (z) in [m], are computed as with as described in Table A.3.5.6.2-3.

Table A.3.5.6.2-3: 4-bit codes and respective room dimension values

|  |  |  |  |
| --- | --- | --- | --- |
| Code | Value | Code | Value |
| 0000 | 0.5 | 1000 | 8 |
| 0001 | 0.707 | 1001 | 11.314 |
| 0010 | 1 | 1010 | 16 |
| 0011 | 1.4141 | 1011 | 22.627 |
| 0100 | 2 | 1100 | 32 |
| 0101 | 2.8282 | 1101 | 45.255 |
| 0110 | 4 | 1110 | 64 |
| 0111 | 5.6568 | 1111 | 90.51 |

The six absorption coefficients, corresponding to the six room surfaces (front, back, left, right, ceiling, floor) are computed as with as shown on Table A.3.5.6.2-4.

Table A.3.5.6.2-4: 2-bit codes and respective absorption coefficient values

|  |  |
| --- | --- |
| Code | Value |
| 00 | 0.0800 |
| 01 | 0.1656 |
| 10 | 0.3430 |
| 11 | 0.7101 |

The most recent 1-byte ACOUSTIC\_ENVIRONMENT PI data (AE identifier) is used to select the acoustic environment for rendering with room acoustics synthesis. This selection stays in effect until a new AE identifier is received.

#### A.3.5.6.3 Audio Description PI data (forward direction)

Audio Description (AD) PI data frames can be used to describe the audio content (e.g. speech/music/general audio) transmitted from sender to receiver.

The size of AD PI data varies from 1 to 5 bytes and depends on the IVAS format as described in Table A.3.5.6.3-1. Wherein for OSBA and OMASA formats the number of discrete coded objects are as per Table 5.8-1 and Table 5.9-1 respectively.

Table A.3.5.6.3-1: Audio Description PI data size

|  |  |
| --- | --- |
| **IVAS format** | **Size (Bytes)** |
| stereo | 1 |
| SBA | 1 |
| MASA | 1 |
| ISM | Number of Objects |
| MC | 2 (1 for center channel + 1 for all other channels) |
| OMASA | 1 + Number of Discrete coded Objects |
| OSBA | 1 + Number of Discrete coded Objects |

Each Byte in AD PI data payload is an audio identifier (AID) that is defined as follows.

AID: An 8 bit identifier, as described in figure A.3.5.6.3-1, to specify type of audio that is being transmitted. This identifier contains V, M, A, E and B field, as defined in Table A.3.5.6.3-2, Table A.3.5.6.3-3, Table A.3.5.6.3-4, Table A.3.5.6.3-5 and Table A.3.5.6.3-6 respectively, which specifies whether audio contains speech, music or ambiance or a combination of these audio types. The E field indicates if the metadata (e.g., orientation, gain, position, direction, etc.) for this audio is editable by the media receiver for rendering. The B field indicates if a stereo stream is binaural or default stereo, i.e., non-binaural. A value of AID where all V, M and A fields equal to 0 corresponds to an unspecified audio type which indicates that an audio description is not available for the related audio frames. The reserved bits in AID shall be set to 0 and be ignored by a receiver.

The latest received AD PI data is used until a new AD PI data is received.

|  |
| --- |
| 0 1 2 3 4 5 6 7  +-+-+-+-+-+-+-+-+  |V|M|A|E|B| RES |  +-+-+-+-+-+-+-+-+ |

Figure A.3.5.6.3-1: Audio Identifier.

Table A.3.5.6.3-2: V field in Audio Identifier Byte

|  |  |
| --- | --- |
| **code** | **value** |
| 0 | Non-speech |
| 1 | speech |

Table A.3.5.6.3-3: M field in Audio Identifier Byte

|  |  |
| --- | --- |
| **code** | **value** |
| 0 | Non-Music |
| 1 | Music |

Table A.3.5.6.3-4: A field in Audio Identifier Byte

|  |  |
| --- | --- |
| **code** | **value** |
| 0 | Absence of background Ambiance |
| 1 | Presence of background Ambiance |

Table A.3.5.6.3-5: E field in Audio Identifier Byte

|  |  |
| --- | --- |
| **code** | **value** |
| 0 | Audio metadata for rendering is not editable |
| 1 | Audio metadata for rendering is editable |

Table A.3.5.6.3-6: B field in Audio Identifier Byte

|  |  |
| --- | --- |
| **code** | **value** |
| 0 | Stereo stream is not binaural (default stereo) |
| 1 | Stereo stream is binaural |

In an example scenario for OSBA format, if the discretely coded objects contain only clean speech then the corresponding Audio Identifier byte should have V field set to 1 and M and A field set to 0 whereas if the coded SBA component contain only background ambiance then the corresponding Audio Identifier byte should have A field set to 1 and V and M field set to 0. Setting E field to 1 indicates that the related audio metadata for rendering may be edited.

#### A.3.5.6.4 ISM specific PI data

##### A.3.5.6.4.1 ISM specific PI data types and structure

ISM related PI data types are listed in table A.3.5.6.4.1-1. Figure A.3.5.6.4.1-1 presents a general PI data structure for ISM related PI types. The ISM PI data may consist of 1-4 blocks of PI data for the corresponding PI type, which together determines the PI size (see Table A.3.5.5-1). In the example presented in Figure A.3.5.6.4.1-1 there are four blocks of two bytes each.

Unless otherwise specified for the PI type, the number of PI data blocks shall be equal to the number of ISMs in the associated IVAS frame. The ISM related PI types can be used for the ISM, OMASA and OSBA coded formats. The latest received ISM related PI data is used until a new ISM related PI data of the same type is received.

|  |
| --- |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | ISM PI data #1 | ISM PI data #2 | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | ISM PI data #3 | ISM PI data #4 | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |

**Figure A.3.5.6.4.1-1: ISM PI data structure with 4 ISM PI data blocks.**

**Table A.3.5.6.4.1-1: ISM related PI data types.**

|  |
| --- |
| **ISM related PI type** |
| ISM\_NUM |
| ISM\_ID |
| ISM\_GAIN |
| ISM\_ORIENTATION |
| ISM\_POSITION |
| ISM\_DISTANCE\_ATTENUATION |
| ISM\_DIRECTIVITY |

##### A.3.5.6.4.2 Number of ISMs

ISM\_NUM PI data specifies the number of ISMs transported in the associcated IVAS frame, as presented in figure A.3.5.6.4.2-1. The first two bits indicate the number of ISMs (N) according to table A.3.5.6.4.2-1. The last six bits are used for zero padding to force byte alignment. The signalled number of ISMs shall be equal to the number of ISMs transported in the associated IVAS frame, and signalled in a single PI data block (i.e. not repeated for each ISM).

To avoid mismatch between the number of ISMs indicated with the ISM\_NUM and the transmitted IVAS frame, ISM\_NUM PI data should be included in the payload when the number of ISMs change between the transmitted IVAS frames.

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

**Figure A.3.5.6.4.2-1: ISM\_NUM PI data frame.**

**Table A.3.5.6.4.2-1: 2-bit codes and the indicated number of ISMs.**

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

##### A.3.5.6.4.3 ISM ID

ISM\_ID PI data specifies an identity (ID) for the ISMs transported in the associated IVAS frame, as presented in figure A.3.5.6.4.3-1. 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 A.3.5.6.4.3-1: ISM\_ID PI data frame.**

##### A.3.5.6.4.4 ISM gain

ISM\_GAIN PI data specifies a gain factor for ISMs transported in the associated IVAS frame, as presented in figure A.3.5.6.4.4-1. 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 7-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 A.3.5.6.4.4-1.

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

**Figure A.3.5.6.4.4-1: ISM\_GAIN PI data frame.**

Table A.3.5.6.4.4-1: 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.3.5.6.4.5 ISM orientation

ISM\_ORIENTATION PI data describes the orientation of the audio object(s) in the ISM(s), with respect to the scene orientation, using orientation data structures in accordance with clause A.3.5.6.1.1. 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 when the number of ISMs N>1.

##### A.3.5.6.4.6 ISM position

ISM\_POSITION PI data indicates the position(s) of audio object(s) in the ISM(s) in the 3D space. Figure A.3.5.7.3-1 shows a general position PI data structure as cartesian coordinates (X, Y, Z) that can be used for ISM\_POSITION PI data frames. The position PI data structures for the transported ISMs are positioned after one another. For example, the ISM position for the first object is followed by the ISM position for the second object when the number of ISMs N>1. See clause A.3.5.7.3 for more information about position PI data.

##### A.3.5.6.4.7 ISM distance attenuation

The ISM\_DISTANCE\_ATTENUATION PI data frame is presented in figure A.3.5.6.4.7-1. 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) per ISM PI data block in accordance with table A.3.5.6.4.7-1 - A.3.5.6.4.7-3. For more information about ISM distance attenuation, see clause 7.2.2.2.7.

The ISM\_DISTANCE\_ATTENUATION PI data frame may consist of a single PI data block, which then is applicable to all ISMs, or separate PI data blocks for each transported ISM, positioned after one another. For example, the ISM\_DISTANCE\_ATTENUATION for the first object is followed by the ISM\_DISTANCE\_ATTENUATION for the second object when the number of ISMs N>1.

|  |
| --- |
| 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.3.5.6.4.7-1: ISM\_DISTANCE\_ATTENUATION PI data frame.**

Table A.3.5.6.4.7-1: 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 A.3.5.6.4.7-2: 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 A.3.5.6.4.7-3: 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.3.5.6.4.8 ISM directivity

ISM\_DIRECTIVITY PI data frame structure is presented in figure A.3.5.6.4.8-1. The ISM orientation indicates 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 A.3.5.6.4.8-1. The outer cone angle, “Outer ang”, (5 bits) determines the size of the outer (back) cone, see table A.3.5.6.4.8-1. 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 A.3.5.6.4.8-2. The total size of an ISM\_DIRECTIVITY PI data frame for a single object is 2 bytes, including zero-padding for byte alignment.

A single PI data block may be used to signal the same ISM directivity for all ISMs, or otherwise ISM directivity PI data blocks for all transported ISMs shall be included and positioned after one another. For example, the ISM directivity for the first object is followed by the ISM directivity for the second object when the number of ISMs N>1. 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.3.5.6.4.8-1: ISM\_DIRECTIVITY PI data frame.**

Table A.3.5.6.4.8-1: 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.3.5.6.4.8-2: 5-bit codes and respective Outer attenuation gain values (dB)

|  |  |  |  |
| --- | --- | --- | --- |
| Code | Value | Code | Value |

|  |  |  |  |
| --- | --- | --- | --- |
| 00000 | -Inf | 10000 | -45 |
| 00001 | -90 | 10001 | -42 |
| 00010 | -87 | 10010 | -39 |
| 00011 | -84 | 10011 | -36 |
| 00100 | -81 | 10100 | -33 |
| 00101 | -78 | 10101 | -30 |
| 00110 | -75 | 10110 | -27 |
| 00111 | -72 | 10111 | -24 |
| 01000 | -69 | 11000 | -21 |
| 01001 | -66 | 11001 | -18 |
| 01010 | -63 | 11010 | -15 |
| 01011 | -60 | 11011 | -12 |
| 01100 | -57 | 11100 | -9 |
| 01101 | -54 | 11101 | -6 |
| 01110 | -51 | 11110 | -3 |
| 01111 | -48 | 11111 | 0 |

##### A.3.5.6.4.9 ISM metadata in IVAS bitstream and PI data

The above described metadata may also be present in the IVAS bitstream. For example, the ISM orientation data may be present in ISM bitstreams of 64 kbps bitrate and above in the extended metadata field. If a specific metadata type for the ISM types (covering also OMASA and OSBA) is present in both the received bitstream and PI data, the metadata in the PI data shall be used. These rules also apply to the diegetic signalling presented in clause A.3.5.6.5.

#### A.3.5.6.5 Diegetic and non-diegetic indication

The DIEGETIC\_TYPE PI type can be used to indicate if the transported audio is diegetic (X=1) or non-diegetic (X=0). The PI data structure is presented in figure A.3.5.6.5-1. The number of bits used for the diegetic/non-diegetic signalling depends on the coded format of the transported audio, see Table A.3.5.6.5-1. The last three bits are used for zero padding to force byte alignment.

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

**Figure A.3.5.6.5-1: DIEGETIC\_TYPE PI data frame.**

Table A.3.5.6.5-1: DIEGETIC\_TYPE bit allocation based on coded format.

|  |  |  |
| --- | --- | --- |
| **TYPE bits** | **Coded format(s)** | **Description** |
| X0000 | Stereo, SBA, MASA, MC | The first bit indicates if the audio is diegetic or non-diegetic. The rest of the TYPE bits are set to zero. |
| XXXX0 | ISM | The first four bits indicate the diegetic/non-diegetic status for each individual ISM (maximum of 4). The last TYPE bit is set to zero. |
| XXXXX | OMASA, OSBA | The first four bits indicate the diegetic/non-diegetic status for each discretely coded individual ISM (maximum of 4). The fifth bit indicates the diegetic/non-diegetic status for the MASA/SBA part of the audio. |

Non-diegetic audio indicates that the headtracking should not be used for the transmitted audio. I.e., the head orientation of the listener should not affect the rendering of the transmitted audio. In case the media receiver does not support headtracked rendering during the time of session negotiation, the DIEGETIC\_TYPE PI type should not be included in the SDP answer from the media receiver (as defined in clause A.4.1).

The DIEGETIC\_TYPE PI type can be used to control the enabling or disabling of headtracking at a media receiver. For example, changing the DIEGETIC\_TYPE from diegetic to non-diegetic can be interpreted as disabling of headtracking at the media receiver, and changing from non-diegetic to diegetic can be interpreted as enabling of headtracking at the media receiver. These are recommendations from the media sender and the media receiver may choose to override these recommendations. 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 clause 7.4.2.2 (Listener orientation). Some other orientations (e.g., scene orientation or device orientation) may be applied at the rendering and those should not be affected by the diegetic/non-diegetic indication.

The DIEGETIC\_TYPE PI type is valid for the current audio frame and for any future audio frames.

The media sender can provide indication to the media receiver 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, DIEGETIC\_TYPE PI data indicating non-diegetic audio created by the sender can be transmitted to the media receiver, if the transmitted spatial audio content already includes orientation changes and no further orientation changes are desired from the headtracking. The DIEGETIC\_TYPE PI data indicating non-diegetic audio 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.3.5.7 Reverse direction PI data types

#### A.3.5.7.1 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 and follows the orientation PI structure presented in clause A.3.5.6.1.1.

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

#### A.3.5.7.2 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 and follows the orientation PI structure presented in clause A.3.5.6.1.1.

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

#### A.3.5.7.3 Listener position

LISTENER\_POSITION PI frame indicates the listener position in 3D space in cartesian coordinates as in figure A.3.5.7.3-1. Figure A.3.5.7.3-1 shows a general position PI data structure as cartesian coordinates (X, Y, Z). Each component is a 16 bits signed integer in units of 0.01 metres. This gives a range of approximately [-327.68, 327.68] meters for each component. 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.

The latest received LISTENER\_POSITION PI data is used until a new LISTENER\_POSITION PI data is received.

|  |
| --- |
| 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.3.5.7.3-1: General position PI data frame with cartesian coordinates.**

#### A.3.5.7.4 Dynamic Audio Suppression

The Dynamic Audio Supression (DAS) PI data describes receiver’s preference with respect to the type of audio content (for e.g., Speech only) that should be enhanced and the amount of suppression to be applied to the background noise, where the background noise is defined as the type of audio content that should be suppressed according to the receiver preference.

The size of DAS PI data is 2 bytes and is described in figure A.3.5.7.4-1. The DAS PI data payload contains an Audio Identifier (AID) byte and a Suppression Level Indicator (SLI) byte, as illustrated in Figure A.3.5.7.4-1. The value of AID field shall be non-zero for the audio identifier bits (V, M, A) and the reserved bits in AID shall be set to 0, unless defined. Likewise, there are 4 useable bits of the SLI while the 4 other (reserved) bits shall be set to 0. The AID byte contains V, M and A field, as defined in Table A.3.5.6.3-2, Table A.3.5.6.3-3 and Table A.3.5.6.3-4 respectively, which specifies whether receiver’s preference is speech, music or ambiance or a combination of these audio types. The Suppression Level Indicator, as defined in table A.3.5.7.4-1, allows specifying a desired degree of suppression where audio signal components other than what is specified by the AID field are considered as undesired audio component. The SLI takes values from 0 to 15 wherein the expected amount of audio suppression is proportional (in approximate logarithmic domain) to the indicator value with 0 indicating no audio suppression and 15 indicating maximum audio suppression as offered by the sender side.

The latest received DAS PI data is used until a new DAS PI data is received.

|  |
| --- |
| 0 1  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |V|M|A| RES | SLI | RES |  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  + AID + SLI + |

Figure A.3.5.7.4-1: DAS PI data byte

Table A.3.5.7.4-1: Suppression Level Indicator

|  |  |  |
| --- | --- | --- |
| **SLI Size bits** | **Value** | **Description** |
| 0000 | 0 | No suppression |
| 0001 | 1 | … |
| … | … | … |
| 1110 | 14 | … |
| 1111 | 15 | Maximum suppression offered by the sender side |

#### A.3.5.7.5 Audio focus direction

AUDIO\_FOCUS\_DIRECTION PI frame indicates a direction of interest from the receiver (i.e., the listener) to the sender side. The focus direction is transmitted similarly as the orientation PI data, see clause A.3.5.6.1.1. The focus direction is relative to the frontal direction (w=0, x=1, y=0, z=0).

The latest received AUDIO\_FOCUS\_DIRECTION PI data is used until a new AUDIO\_FOCUS\_DIRECTION PI data is received.

#### A.3.5.7.6 Reverse PI latency

A receiving 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 or as part of the IVAS bit stream. 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 endpoints to e.g. send better predicted orientations as reverse PI data or to apply better predicted orientation data. This PI latency is sent back the sending device using the PI\_LATENCY PI frames. Figure A.3.5.7.6-1 shows the data structure of a PI\_LATENCY frame, which consists of the 5-bit type of the reverse PI frame according to table A.3.5.5-1A used for latency estimation and a signed 27-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.

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

|  |
| --- |
| 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  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  | type | PI latency |  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |

Figure A.3.5.7.6-1: PI latency indication.

#### A.3.5.7.7 ISM specific PI data editing requests

##### A.3.5.7.7.1 General

Clauses A.3.5.7.7.2 – A.3.5.7.7.6 present editing request PI types for ISMs. The PI types can be used to transmit spatial audio editing information requests for ISMs to a media sender. The media sender can use the spatial audio editing information to modify the input spatial audio signal accordingly and encode and transmit the modified audio to the media receiver for decoding and rendering. The editing request PI types can be used to perform object editing at the media sender side.

If present, there shall be at most one editing request for a single ISM and only a single ISM shall be requested to be edited in an RTP payload. The ISM to be edited is identified with a R\_ISM\_ID identifier which contains the identifier information for a received ISM stream. The editing requests can be used for the ISM, OMASA and OSBA coded formats.

The latest received editing request PI data for the identified ISM is used until a new editing request for the same ISM is received.

##### A.3.5.7.7.2 ISM ID in editing requests

R\_ISM\_ID PI data specifies an identity (ID) of a received ISM that is requested to be edited with ISM specific PI data request in this payload. The ID data frame follows the structure presented in figure A.3.5.6.4.3-1. The PI data includes an identity field (one byte) for a single received ISM requested to be edited.

##### A.3.5.7.7.3 Editing request for ISM gain

R\_ISM\_GAIN PI data specifies a gain factor editing request for a received ISM. The gain factor editing request data frame follows the structure presented in figure A.3.5.6.4.4-1. The 7-bit ISM gains represent a uniform range (in dB) ranging from -96 dB to +3 dB with additional code point (-Inf dB) for requesting to mute an audio object, according to table A.3.5.6.4.4-1.

##### A.3.5.7.7.4 Editing request for ISM orientation

R\_ISM\_ORIENTATION PI data describes an orientation editing request for a received ISM, with respect to the scene orientation, using orientation data structures in accordance with clause A.3.5.6.1.1.

##### A.3.5.7.7.5 Editing request for ISM position

R\_ISM\_POSITION PI data describes a positional editing request in the 3D space for a received ISM. Figure A.3.5.7.3-1 shows a general position PI data structure as cartesian coordinates (X, Y, Z) that can be used for R\_ISM\_POSITION PI data frames. See clause A.3.5.7.3 for more information about position PI data.

##### A.3.5.7.7.6 Editing request for ISM direction

R\_ISM\_DIRECTION PI data describes a direction editing request for a received ISM. The data frame follows the structure presented in figure A.3.5.7.7.8-1 where the azimuth angle component has 9 bits reserved for the value and the elevation component has 7 bits reserved for the value. The azimuth angles range from -180° (exclusive) to 180° with approximately 0.70° step size and the elevation angles range from -90° (inclusive) to 90° with approximately 1.41° step size. The codes and the respective azimuth and elevation values are presented in tables A.3.5.7.7.8-1 and A.3.5.7.7.8-2.

|  |
| --- |
| 0 1  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | azi | elev | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |

**Figure A.3.5.7.7.8-1: R\_ISM\_DIRECTION PI data frame.**

Table A.3.5.7.7.8-1: 9-bit codes and respective (approximate) azimuth values (degrees)

|  |  |
| --- | --- |
| Code | Value |
| 0 0000 0000 | -179.30° |
| 0 0000 0001 | -178.60° |
| 0 0000 0010 | -177.90° |
| 0 0000 0011 | -177.19° |
| 0 0000 0100 | -176.48° |
| 0 0000 0101 | -175.78° |
|  |  |
| 1 1111 1100 | 177.90° |
| 1 1111 1101 | 178.60° |
| 1 1111 1110 | 179.30° |
| 1 1111 1111 | 180° |

Table A.3.5.7.7.8-2: 7-bit codes and respective (approximate) elevation values (degrees)

|  |  |
| --- | --- |
| Code | Value |
| 000 0000 | -90° |
| 000 0001 | -88.59° |
| 000 0010 | -87.17° |
| 000 0011 | -85.76° |
| 000 0100 | -84.34° |
| 000 0101 | -82.93° |
|  |  |
| 111 1100 | 85.76° |
| 111 1101 | 87.17° |
| 111 1110 | 88.59° |
| 111 1111 | 90° |

# 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.

Media type name: audio

Media subtype name: IVAS

Required parameters: none

Optional parameters:

The parameters defined below apply to RTP transfer only:

**ptime**: see [32].

**maxptime**: see [32].

**dtx/dtx-recv**: as defined in Annex A of [3].

**max-red**: see [36].

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

NOTE: The use of the channels parameter as defined in [35] does not permit signaling all IVAS Immersive mode coded formats; formats need to be derived from the cf/cf-send/cf-recv parameters.

**mono-init**: This parameter defines the mode at the start or update of the session for the direction specified by the session directionality attribute or the suffix. Permissible values are 0 and 1. If mono-init is 0 or not present, IVAS Immersive mode is used and the coded format is defined by the cf parameter. If mono-init is 1, the coded format is the EVS-compatible mono mode of IVAS; 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.

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

NOTE: The evs-mode-switch parameter only applies to the direction for which the mono-init parameter is 1.

**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.

**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 ibr 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 this parameters is not present and not otherwise specified by ibr-send or ibr-recv, 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 this parameter is not present and not otherwise specified by ibw-send or ibw-recv, 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.4.1-1 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.4.1-1: 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 payloads are self-contained for all IVAS coded formats except SR and mono, i.e., they require no additional signaling for decoding than the payload size.

Table A.4.1-2: List of coded subformats for all coded formats except Stereo and SR

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Main format | List of allowed subformats | cf-sub values |  | Main format | List of allowed subformats | cf-sub values |
| cf=SBA | FOA planar | FOA\_P | cf=OMASA | OMASA ISM1 1TC | ISM1\_MASA\_1TC |
| HOA2 planar | HOA2\_P | OMASA ISM2 1TC | ISM2\_MASA\_1TC |
| HOA3 planar | HOA3\_P | OMASA ISM3 1TC | ISM3\_MASA\_1TC |
| FOA | FOA | OMASA ISM4 1TC | ISM4\_MASA\_1TC |
| HOA2 | HOA2 | OMASA ISM1 2TC | ISM1\_MASA\_2TC |
| HOA3 | HOA3 | OMASA ISM2 2TC | ISM2\_MASA\_2TC |
| cf=MASA | MASA1 | MASA1 | OMASA ISM3 2TC | ISM3\_MASA\_2TC |
| MASA2 | MASA2 | OMASA ISM4 2TC | ISM4\_MASA\_2TC |
| cf=ISM | ISM1 | ISM1 | cf=OSBA | OSBA ISM1 FOA planar | ISM1\_FOA\_P |
| ISM2 | ISM2 | OSBA ISM2 FOA planar | ISM2\_FOA\_P |
| ISM3 | ISM3 | OSBA ISM3 FOA planar | ISM3\_FOA\_P |
| ISM4 | ISM4 | OSBA ISM4 FOA planar | ISM4\_FOA\_P |
| ISM1 extended metadata | ISM1\_ext | OSBA ISM1 FOA | ISM1\_FOA |
| ISM2 extended metadata | ISM2\_ext | OSBA ISM2 FOA | ISM2\_FOA |
| ISM3 extended metadata | ISM3\_ext | OSBA ISM3 FOA | ISM3\_FOA |
| ISM4 extended metadata | ISM4\_ext | OSBA ISM4 FOA | ISM4\_FOA |
| cf=MC | MC 5.1 | 5\_1 | OSBA ISM1 HOA2 planar | ISM1\_HOA2\_P |
| MC 7.1 | 7\_1 | OSBA ISM2 HOA2 planar | ISM2\_HOA2\_P |
| MC 5.1.2 | 5\_1\_2 | OSBA ISM3 HOA2 planar | ISM3\_HOA2\_P |
| MC 5.1.4 | 5\_1\_4 | OSBA ISM4 HOA2 planar | ISM4\_HOA2\_P |
| MC 7.1.4 | 7\_1\_4 | OSBA ISM1 HOA2 | ISM1\_HOA2 |
|  | | | OSBA ISM2 HOA2 | ISM2\_HOA2 |
| OSBA ISM3 HOA2 | ISM3\_HOA2 |
| OSBA ISM4 HOA2 | ISM4\_HOA2 |
| OSBA ISM1 HOA3 planar | ISM1\_HOA3\_P |
| OSBA ISM2 HOA3 planar | ISM2\_HOA3\_P |
| OSBA ISM3 HOA3 planar | ISM3\_HOA3\_P |
| OSBA ISM4 HOA3 planar | ISM4\_HOA3\_P |
| OSBA ISM1 HOA3 | ISM1\_HOA3 |
| OSBA ISM2 HOA3 | ISM2\_HOA3 |
| OSBA ISM3 HOA3 | ISM3\_HOA3 |
| OSBA ISM4 HOA3 | ISM4\_HOA3 |
| NOTE: No subformats exist for Stereo and SR. | | | | | | |

**cf-send/cf-recv**: cf parameter in send or receive direction. If the cf-recv parameter is not present and not otherwise specified by cf, all IVAS coded formats consistent with the negotiated bitrate(s) are allowed in the session in receive direction.

**cf-sub-info**: Specifies the IVAS Immersive mode subformats supported in the send direction for each of the coded formats included in the cf or cf-send parameter. Permissible values of subformats for each coded format are listed in the Table A.4.1-2. The subformats listed in the cf-sub-info parameter restrict the available subformats for requests in clause A.3.3.3.3.3.3.

**ivas-icm**: IVAS initial codec mode (ivas-icm) parameter contains the information required by a media receiver to initialize the decoder, renderer and playout based on the media sender configuration at the start or update of the session. The parameter contains the subformat (according to Table A.4.1-2), bitrate (in kilobits per second) and bandwidth (wb, swb or fb) used at the start of the session in a colon separated list for each coded format listed in the cf parameter. Since Stereo and SR coded formats do not have subformats, the subformat placeholder in the ivas-icm parameter uses ‘Stereo’ and ‘SR’. The ivas-icm parameter list for each coded format is separated with a comma (e.g., sub1:ibr1:ibw1,sub2:ibr2:ibw2). A media receiver may use this information to initialize the media receiver before any audio frame is received, e.g. to decrease the startup latency or to avoid computational complexity peaks. If the parameter is empty or not present, the media receiver cannot assume the behavior of the media sender at the start or update of the session.

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

NOTE: In case some RTP packets arrive before the SDP offer/answer settles down, the media receiver determines the initialization information based on the audio frames from the RTP stream regardless of the cf or ivas-icm parameters in the SDP.

**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.3.5.5-1, A.3.5.5-1A and A.3.5.5-2. If the pi-types parameter is not present and not otherwise specified by pi-types-send or pi-types-recv, 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 peak bitrate for the PI data section (excluding the E-bytes for indication) for each packet in the session in kilobits per second. Bitrate calculation for PI data shall take the packet interval, i.e. value of ptime into account. The parameter indicates the maximum bitrate for the PI data. If pi-br parameter is not present and not otherwise specified by pi-br-send or pi-br-recv, a default value of 0 shall be used.

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

**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 pose corrections around a single axis only, D=2 means support of pose corrections around a two axes , D=3 means support of pose corrections around all 3 axes. If the first value is greater than -1, it 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. A positive value of D may additionally be appended with the optional suffix \*, indicating high-efficiency rather than high-quality split renderer metadata calculation to be used in the session.

**sr-tc:** Specifies the codec format for the binaural transport channels. This parameter must be present in a negotiated split rendering session. Permissible values are ‘LCLD’ and ‘LC3plus’ followed by a an optional value specifying the maximum allowed split rendering bitrate, in kilobits per second, for the SR stream(s) to be used in the session. If the bit rate value is left empty, the maximum of 512 is assumed. Table A.3.3.3.2-2 specifies the bitrates that can be specified as maximum bitrates. All value fields shall be separated by commas even if left empty.

**sr-tc-fr:** Specifies the split rendering transport channel codec frame size. This parameter is only applicable for non-diegetic streams or streams without pose correction metadata (D=-1 or D=0 in sr-dof parameter). Allowed values for are 5, 10, 20 (corresponding to ms). If sr-tc-fr is not present in a negotiated SR session, a default value of 20 shall be used. If present for diegetic streams or streams with pose correction metadata (D>0 in sr-dof parameter), the parameter shall be ignored and the default value of 20 shall be used.

**sr-tc-cp:** Specifies the additional parameters for split rendering transport channel codec. This parameter is only applicable for LC3plus coded streams. Two comma-separated values corresponding to ‘fdi’ and ‘bwr’ shall be supplied. ‘fdi’ is specified in [Ref TS 103 634]; it shall be set to "1" or "2", depending on the split rendering configuration. ‘bwr’ is specified in [Ref TS 103 634]; it shall be set to "fb".

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 mono-init 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]. If this parameter is not present and the ibr parameter is present, then the limits of the ibr parameter apply also to this parameter if within the allowed range of the br parameter. Otherwise the default limits as defined in Annex A of [3] apply.

**br-send**: as defined in Annex A of [3]. If this parameter is not present and the ibr-send parameter is present, then the limits of the ibr-send parameter apply also to this parameter if within the allowed range of the br-send parameter. Otherwise the default limits as defined in Annex A of [3] apply.

**br-recv**: as defined in Annex A of [3]. If this parameter is not present and the ibr-recv parameter is present, then the limits of the ibr-recv parameter apply also to this parameter if within the allowed range of the br-recv parameter. Otherwise the default limits as defined in Annex A of [3] apply.

**bw**: as defined in Annex A of [3]. If this parameter is not present and the ibw parameter is present, then the limits of the ibw parameter apply also to this parameter if within the allowed range of the bw parameter. Otherwise the default limits as defined in Annex A of [3] apply.

NOTE: Narrow-band is not supported for IVAS operation

**bw-send**: as defined in Annex A of [3]. If this parameter is not present and the ibw-send parameter is present, then the limits of the ibw-send parameter apply also to this parameter if within the allowed range of the ibw-send parameter. Otherwise the default limits as defined in Annex A of [3] apply.

**bw-recv**: as defined in Annex A of [3]. If this parameter is not present and the ibw-recv parameter is present, then the limits of the ibw-recv parameter applies also to this parameter if within the allowed range of the bw-recv parameter. Otherwise the default limits as defined in Annex A of [3] apply.

**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 [36].

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

**mode-change-neighbor**: see [36].

## 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) [32], 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 [33].

## 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:

**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.

**mono-init**: When the mono-init is defined for the send and the receive directions, mono-init should be used but mono-init-send and mono-init-recv may also be used. mono-init can be used even if the session is negotiated to be sendonly, recvonly, or inactive. For sendonly session, mono-init and mono-init-send can be interchangeably used. For recvonly session, mono-init and mono-init-recv can be interchangeably used. When mono-init is not offered for a payload type, the answerer may include mono-init for the payload type in the SDP answer. When mono-init is offered for a payload type and the payload type is accepted, the answerer shall not modify or remove mono-init for the payload type in the SDP answer.

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

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

NOTE: The IVAS codec does not support switching of coded formats (see Table A.4.1-1) without reinitialization. Change of formats would therefore require reinitialization handling for the IVAS codec on application level.

**cf-send**: The SDP offer shall contain the cf-send parameter and 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 in cf-recv or 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 or update of a session and may only be modified by the adaptation mechanisms present in this specification. 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. If cf-recv is not offered for a payload type, cf-send in the answer may indicate any coded format.

**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.

**cf-sub-info**: If present, the parameter lists supported subformats for each coded format in the cf parameter having a defined subformat according to table A.4.1-2 for the session. The cf-sub-info is a declarative parameter, and the parameter is not mirrored in the SDP answer. The media receiver may include their own cf-sub-info in the SDP answer.

**ivas-icm**: If present, the parameter shall list the subformat, bitrate and bandwidth used by the media sender at the start or update of the session in a colon separated list for each coded format in the SDP offer. In case there are multiple coded formats in the SDP offer, the ivas-icm parameter list for each of the coded format is carried as a comma separated list in the same order as the coded formats in the cf parameter (e.g., sub1:ibr1:ibw1,sub2:ibr2:ibw2). The listed parameter values shall comply with the relevant parameters (cf or cf-sub-info, ibr, ibw) in the SDP offer. The ivas-icm parameter list corresponding to only the first coded format listed in the cf parameter in the SDP answer shall be included in the SDP answer. If the SDP answer modifies the bitrate and/or bandwidth range the media receiver may lower the bitrate and/or bandwidth listed in the ivas-icm parameter in the SDP answer. When the same ivas-icm parameter values are defined for the send and the receive directions, ivas-icm should be used but ivas-icm-send and ivas-icm-recv may also be used. For sendonly session, ivas-icm and ivas-icm-send can be interchangeably used. For recvonly session, ivas-icm (or the directional variants) shall not be used.

**ivas-icm-send**: When ivas-icm-send is offered for a payload type and the payload is accepted, the answerer shall include ivas-icm-recv in the SDP answer, and the ivas-icm-recv shall be identical to or a subset of ivas-icm-send for the payload type in the SDP offer with the exception that the listed bitrate and bandwidth values may be lowered.

**ivas-icm-recv**: The ivas-icm-recv parameter shall not be present in the initial SDP offer. The ivas-icm-recv shall be present in the SDP answer only if ivas-icm or ivas-icm-send is present in the initial 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. The pi types listed in the SDP answer should be supported and applicable for the session. 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.

Split rendering related offer-answer considerations

**cf-recv:** When cf-recv is offered for a payload type (typically by lightweight end device), it must list at least SR as one IVAS Immersive mode coded formats. To accept the offer with split rendering used in the subsequent session, the answer shall contain cf\_send with SR as the only IVAS Immersive mode coded format.

**cf-send:** When cf-send is offered for a payload type (typically by a pre-rendering node/device other than lightweight end device), it must list at least SR as one IVAS Immersive mode coded formats. To accept the offer with split rendering used in the subsequent session, the answer shall contain cf\_recv with SR as the only IVAS Immersive mode coded format.

**sr-dof**: When cf-recv or cf-send is offered for a payload type with SR listed, the offer may additionally contain the parameter sr-dof. In that case and if the SR session is accepted, the answerer shall include sr-dof in the SDP answer, and the sr-dof shall be identical or lower than sr-dof in the SDP offer. If the first value in the answer is reduced to -1, the second value shall not be present. The answerer may add but shall not remove a \* suffix unless the value is smaller than 1.

**sr-tc**: When cf-recv or cf-send is offered for a payload type with SR listed, the offer may additionally contain the parameter sr-tc. If the SR session is accepted, the answerer shall include sr-tc in the SDP answer. Unless the SDP offer was open, the sr-tc parameter in the answer shall list all the codecs or a subset of codecs that were present in the the offer. If the SDP offer was open, the sr-tc parameter in the answer shall list at least one codec. If a bitrate value was specified in the offer, the same or a lower bitrate value out of the set of available bitrates shall be used in the answer. If not present, the answer may leave that field open or specify a bitrate.

**sr-tc-fr**: When cf-recv or cf-send is offered for a payload type with SR listed, the offer may additionally contain the parameter sr-tc-fr. In that case and if the SR session is accepted, the answerer shall include sr-tc-fr in the SDP answer, and the sr-tc-fr parameter shall be identical to the sr-tc-fr in the SDP offer.

**sr-tc-cp**: When cf-recv or cf-send is offered for a payload type with SR listed and LC3plus listed in the sr-tc offer then the offer shall additionally contain the parameter sr-tc-cp. If the SR session is accepted with LC3plus listed in the sr-tc answer, the answer shall additionally include the parameter sr-tc-cp. Unless the SDP offer was open, the sr-tc-cp parameter shall be identical to the sr-tc-cp in the SDP offer.

NOTE: As split rendering sessions are typically limited to one direction between two directly connected nodes/end points of an IVAS codec session, only directional parameters shall be used to negotiate an IVAS session with split rendering on that connection. When a SR session is accepted, the SDP answer shall not contain any other session parameters for the direction towards the lightweight end device than the split rendering related parameters listed above, PI data related parameters and cmr. The other direction (from the lightweight device) remains unconstrained with the only exception that directional parameters shall be used to specify the session on it.

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