BLUETOOTH DOC	Date / Year-Month-Day	Approved	Revision	Document No
	2003-05-22	Version	1.0	
Prepared	e-mail address			N.B.
Bluetooth Audio Video Working Group	avv-feedback@bluetooth.org			

ADVANCED AUDIO DISTRIBUTION PROFILE SPECIFICATION

Adopted version 1.0

Abstract

This profile defines the requirements for Bluetooth™ devices necessary for support of the high quality audio distribution. The requirements are expressed in terms of end-user services, and by defining the features and procedures that are required for interoperability between Bluetooth devices in the Audio Distribution usage model.

Revision History

Revision	Date	Comments
0.5	April 2001	Release to Associates
0.7	June 2001	Release to Associates
0.9	September 2001	Release to Associates and Early Adopters
Voting Draft 0.95	October 2001	Release to Associates and Early Adopters
0.95b	March 2002	Adopted 0.95
Voting Draft 1.00	May 2002	Release for Voting Draft
Voting Draft 1.00 a	June 2002	Release for Voting Draft
Voting Draft 1.00 b	February 2003	Release for Voting Draft
Version 1.0	May 2003	Adopted version

Contributors

Morgan Lindqvist	Ericsson
Fisseha Mekuria	Ericsson
Tsuyoshi Okada	Matsushita Electric Industrial
Kalervo Kontola	Nokia
Vesa Lunden	Nokia
Jurgen Schnitzler	Nokia
Shaun Barrett	Philips
Christian Bouffioux	Philips
Frans de Bont	Philips
Rob J. Davies	Philips
Emmanuel Mellery	Philips
Marc Vauclair	Philips
Kenzo Akagiri	Sony
Masakazu Hattori	Sony
Harumi Kawamura (Owner)	Sony
Rudiger Mosig	Sony
Yoshiyuki Nezu	Sony
Masayuki Nishiguchi	Sony
Hiroyasu Noguchi (Co-Owner)	Sony
Tomoko Tanaka	Sony
Junko Ami	Toshiba
Takeshi Saito	Toshiba
Yoshiaki Takabatake	Toshiba
Yoichi Takebayashi	Toshiba
Ichiro Tomoda	Toshiba
Junichi Yoshizawa	Toshiba

Disclaimer and Copyright Notice

The copyright in these specifications is owned by the Promoter Members of Bluetooth SIG, Inc. (“Bluetooth SIG”). Use of these specifications and any related intellectual property (collectively, the “Specification”), is governed by the Promoters Membership Agreement among the Promoter Members and Bluetooth SIG (the “Promoters Agreement”), certain membership agreements between Bluetooth SIG and its Adopter and Associate Members (the Membership Agreements”) and the Bluetooth Specification Early Adopters Agreements (“1.2 Early Adopters Agreements”) among Early Adopter members of the unincorporated Bluetooth special interest group and the Promoter Members (the “Early Adopters Agreement”). Certain rights and obligations of the Promoter Members under the Early Adopters Agreements have been assigned to Bluetooth SIG by the Promoter Members.

Use of the Specification by anyone who is not a member of Bluetooth SIG or a party to an Early Adopters Agreement (each such person or party, a “Member”), is prohibited. The legal rights and obligations of each Member are governed by their applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement. No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

Any use of the Specification not in compliance with the terms of the applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement is prohibited and any such prohibited use may result in termination of the applicable Membership Agreement or Early Adopters Agreement and other liability permitted by the applicable agreement or by applicable law to Bluetooth SIG or any of its members for patent, copyright and/or trademark infringement.

THE SPECIFICATION IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, SATISFACTORY QUALITY, OR REASONABLE SKILL OR CARE, OR ANY WARRANTY ARISING OUT OF ANY COURSE OF DEALING, USAGE, TRADE PRACTICE, PROPOSAL, SPECIFICATION OR SAMPLE.

Each Member hereby acknowledges that products equipped with the Bluetooth™ technology (“Bluetooth™ Products”) may be subject to various regulatory controls under the laws and regulations of various governments worldwide. Such laws and regulatory controls may govern, among other things, the combi-nation, operation, use, implementation and distribution of Bluetooth™ Products. Examples of such laws and regulatory controls include, but are not limited to, airline regulatory controls, telecommunications regulations, technology transfer controls and health and safety regulations. Each Member is solely responsible for the compliance by their Bluetooth™ Products with any such laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their Bluetooth™ Products related to such regulations within the applicable jurisdictions. Each Member acknowledges that nothing in the Specification provides any information or assistance in connection with securing such compliance, authorizations or licenses. NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS.

ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS OR FOR NONCOMPLIANCE WITH LAWS, RELATING TO USE OF THE SPECIFICATION IS EXPRESSLY DISCLAIMED. BY USE OF THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST BLUETOOTH SIG AND ITS PROMOTER MEMBERS RELATED TO USE OF THE SPECIFICATION.

Bluetooth SIG reserves the right to adopt any changes or alterations to the Specification as it deems necessary or appropriate and to adopt a process for adding new Bluetooth™ profiles after the release of the Specification.

Document Terminology

The Bluetooth SIG has adopted Section 13.1 of the IEEE Standards Style Manual, which dictates use of the words ``shall'', ``should'', ``may'', and ``can'' in the development of documentation, as follows:

- The word *shall* is used to indicate mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).

- The use of the word *must* is deprecated and shall not be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

- The use of the word *will* is deprecated and shall not be used when stating mandatory requirements; *will* is only used in statements of fact.

- The word *should* is used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain course of action is deprecated but not prohibited (*should* equals *is recommended that*).

- The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted*).

- The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

Contents

1 Introduction

1.1 Scope

The Advanced Audio Distribution Profile (A2DP) defines the protocols and procedures that realize distribution of audio content of high-quality in mono or stereo on ACL channels. The term “advanced audio”, therefore, should be distinguished from “Bluetooth audio”, which indicates distribution of narrow band voice on SCO channels as defined in Chapter 12 of Bluetooth Baseband specification [1].

A typical usage case is the streaming of music content from a stereo music player to headphones or speakers. The audio data is compressed in a proper format for efficient use of the limited bandwidth. Surround sound distribution is not included in the scope of this profile.

The A2DP focuses on audio streaming, while the Video Distribution Profile (VDP) specifies video streaming. Support of both profiles enables us to distribute video content accompanied with high-quality audio. The usage case of video and audio streaming is described in the VDP.

Note also that the A2DP does not include remote control functions. Devices may support remote control features by implementing both A2DP and the control profile as depicted, for example, in the usage scenario of Audio/Video Remote Control Profile[2].

Editor's note: The A/V WG has requested additional QoS support in the next revision of the Bluetooth data link specification. When other profiles with stringent requirements are used in conjunction with this profile the performance may be degraded due to insufficient support of QoS in the current Bluetooth specification v.1.1, which all profiles use.

1.2 Profile Dependency

In Figure 1.1, the structure and the dependencies of the profiles are depicted. A profile is dependent upon another profile if it re-uses parts of that profile, by implicitly or explicitly referencing it. Dependency is illustrated in the figure. A profile has dependencies on the profile(s) in which it is contained – directly and indirectly.

As indicated in the figure, the A2DP is dependent upon the Generic Access Profile (GAP), and also the Generic Audio/Video Distribution Profile (GAVDP) [3], which defines procedures required to setup an audio/video streaming. The A2DP defines parameters and procedures that are specific for audio streaming. The terminology, user interface and procedures as defined in the GAP and GAVDP are applicable to this profile, unless explicitly stated otherwise.

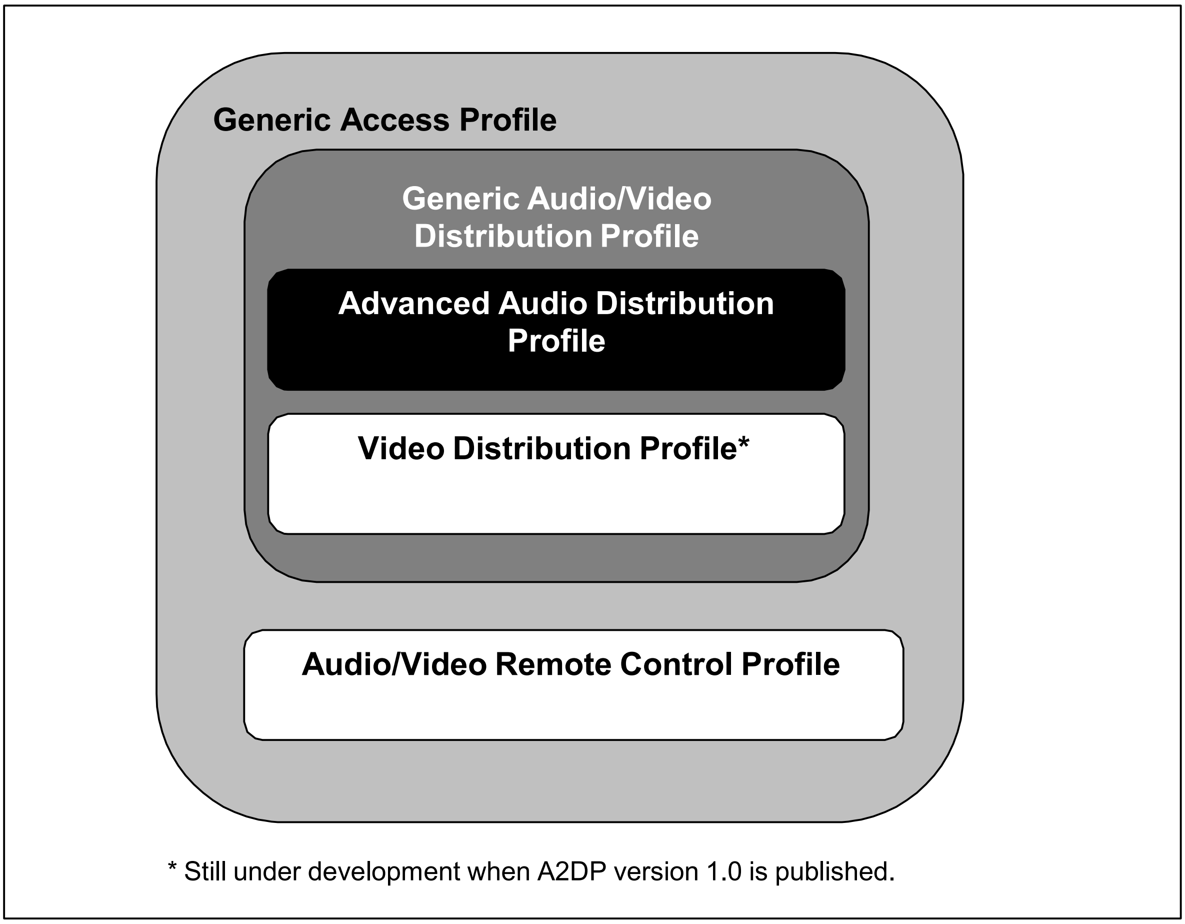

Generic Access Profile
Generic Audio/Video Distribution Profile
Advanced Audio Distribution Profile
Video Distribution Profile*
Audio/Video Remote Control Profile
* Still under development when A2DP version 1.0 is published.

Figure 1.1: Profile Dependencies

1.3 Symbols and Conventions

1.3.1 Requirement Status Symbols

In this document the following symbols are used:

‘M’ for mandatory to support (used for capabilities that shall be used in the profile).

‘O’ for optional to support (used for capabilities that may be used in the profile).

‘C’ for conditional support (used for capabilities that shall be used in case a certain other capability is supported).

‘X’ for excluded (used for capabilities that may be supported by the unit, but that shall never be used in the profile).

‘N/A’ for not applicable (in the given context it is impossible to use this capability).

Some excluded capabilities are capabilities that, according to the relevant Bluetooth specification, are mandatory. These are features that may degrade operation of devices following this profile. Therefore, these features shall never be activated while a unit is operating as a unit within this profile.

1.3.2 Definition

1.3.2.1 RFA

Reserved for Future Additions. Bits with this designation shall be set to zero. Receivers shall ignore these bits.

1.3.2.2 RFD

Reserved for Future Definition. These bit value combinations or bit values are not allowed in the current specification but may be used in future versions. The receiver shall check that unsupported bit value combination is not used.

2 Profile Overview

2.1 Profile Stacks

The figure below shows the protocols and entities used in this profile.

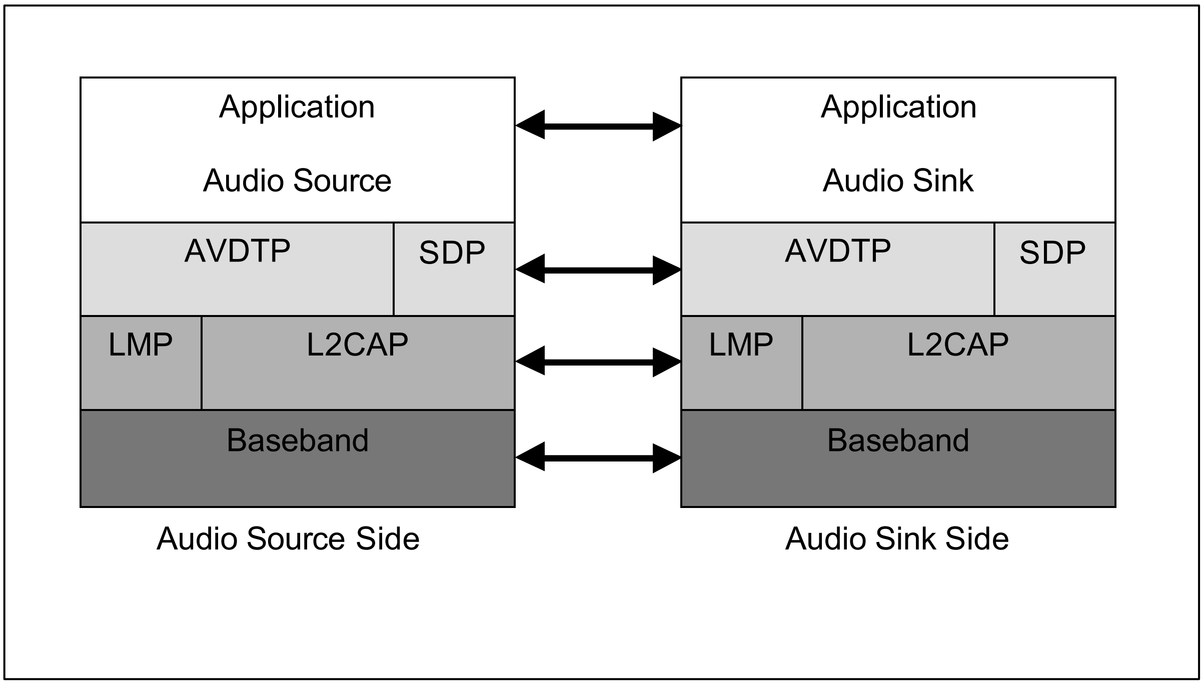


Figure 2.1: Protocol Model

The Baseband[1], LMP[5], L2CAP[6], and SDP[7] are Bluetooth protocols defined in the Bluetooth Core specifications. AVDTP[4] consists of a signalling entity for negotiation of streaming parameters and a transport entity that handles streaming itself.

The Application layer shown in Figure 2.1 is the entity in which the device defines application service and transport service parameters. The entity also adapts the audio streaming data into the defined packet format, or vice versa.

For the shaded protocols/entities in Figure 2.1, the GAVDP applies, except in those cases where this profile explicitly states deviations.

2.2 Configurations and Roles

The following roles are defined for devices that implement this profile:

Source (SRC) – A device is the **SRC** when it acts as a source of a digital audio stream that is delivered to the **SNK** of the piconet.

Sink (SNK) – A device is the **SNK** when it acts as a sink of a digital audio stream delivered from the **SRC** on the same piconet.

Examples of configurations illustrating the roles for this profile are depicted in Figure 2.2.

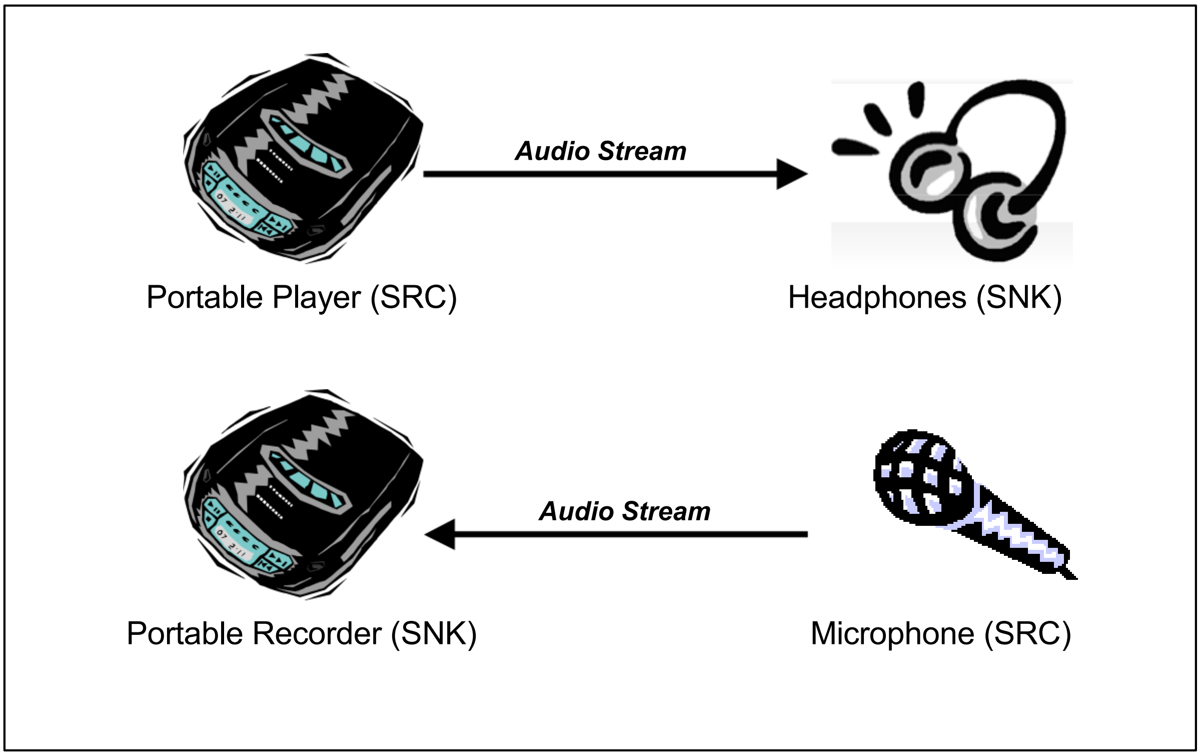


Figure 2.2: Examples of Configuration

2.3 User Requirements and Scenarios

The following scenario is covered by this profile:

- Setup/control/manipulate a streaming of audio data from the **SRC** to the **SNK**(s).

The following restrictions are applied to this profile:

1 The profile does not support a synchronized point-to-multipoint distribution.

2 There exists certain delay between the **SRC** and the **SNK** due to radio signal processing, data buffering, and encode/decode of the stream data. Countering the effects of such delays depends on implementation.

The following requirements are set in this profile:

3 The audio data rate should be sufficiently smaller than usable bit rate on the Bluetooth link. This is to allow retransmission schemes to reduce the effects of packet loss.

4 The profile does not exclude any content protection method.

2.4 Profile Fundamentals

The profile fundamentals are same as defined in the GAVDP in addition to the following requirement.

- Content Protection is provided at the application level and is not a function of the Bluetooth link level security protocol.

2.5 Conformance

When conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner (process mandatory). This also applies for optional and conditional capabilities for which support is indicated. All mandatory, optional, and conditional capabilities, for which support is indicated, are subject to verification as part of the Bluetooth certification program.

3 Application Layer

This section describes the feature requirements on units complying with the A2DP.

Table 3.1 shows the feature requirements for this profile.

Item No.	Feature	Support in SRC	Support in SNK
1	Audio Streaming	M	M

Table 3.1: Application Layer Features

Table 3.2 maps each feature to the procedures used for that feature, and shows whether the procedure is optional, mandatory, or conditional. The procedures are described in the reference section.

Item No.	Feature	Procedure	Ref.	Support in SRC	Support in SNK
1	Audio Streaming	Send Audio Stream	3.2.1	M	N/A
		Receive Audio Stream	3.2.2	N/A	M

Table 3.2: Application Layer Feature to Procedure Mapping

3.1 Audio Streaming Set Up

When a device wishes to start streaming of audio content, the device firstly needs to set up a streaming connection. Signalling procedures and typical signalling flows are illustrated in Section 4.1 and Appendix A of GAVDP[3], respectively. During such set up procedure, the devices select the most suitable audio streaming parameters. There are two kinds of services configured; one is an application service capability, and the other is a transport service capability. (For details, see Section 4.4 in AVDTP[4].) This profile specifies audio-specific parameters necessary for these signalling procedures. An example of how the session signalling is performed is described in Chapter 14 of GAVDP[3] and in Chapter 13 of this specification.

The application service capability for A2DP consists of audio codec capability and content protection capability. Requirements for audio codec interoperability and details of codec parameters such as mode, sampling frequency, and bit rate are described in Chapter 4. The content protection capability is described in Appendix A as informative.

The transport service capability is provided by AVDTP in order to manipulate the streaming packets more intelligently. Appropriate configuration of these services increases channel throughput. Available services are listed in Section 5.1.2.

3.2 Audio Streaming

Once streaming connection is established and *Start Streaming* procedure in GAVDP is executed, both **SRC** and **SNK** are in the STREAMING state, in which the **SRC**

(**SNK**) is ready to send (receive) audio stream. (See Section 4.1 in GAVDP[3].) The **SRC** uses the *Send Audio Stream* procedure to send audio data to the **SNK**, which in turn employs the *Receive Audio Stream* procedure to receive the audio data. The block diagrams of these procedures and created packet format are shown in Figure 3.1. In Chapter 4 audio-specific parameters in AVDTP header and media payload format are also specified.

Note again that the devices shall be in the STREAMING state to send/receive audio stream. If the **SRC**/**SNK** wishes to send/receive the audio stream whereas the state is still at OPEN, the **SRC**/**SNK** shall initiate *Start Streaming* procedure defined in GAVDP.

3.2.1 Send Audio Stream

In the *Send Audio Stream* procedure, the **SRC** shall, if needed, encode the data into a selected format in the signalling session. Then, the application layer of the **SRC** shall adapt the encoded data into the defined media payload format. The frame of encoded audio data is adapted to the defined payload format as defined in Chapter 4.

When content protection is in use, a content protection header may precede encrypted audio content. This is content protection method dependent.

Afterwards, the stream data shall be handed down to the AVDTP entity through the exposed interface (Interface 4) defined in Chapter 2 of AVDTP[4]. The stream data shall be sent out on the transport channel using the selected transport services defined in Section 5.4 of AVDTP[4].

3.2.2 Receive Audio Stream

The AVDTP entity of the **SNK** shall receive the stream data from the transport channel using the selected transport services and pass it to the application layer by exposed interface defined in Chapter 2 of AVDTP[4].

When a content protection method is active, the application layer of the **SNK** shall process the retrieved AVDTP payload as described by the content protection method. Typically, this processing entails content protection header analysis and decryption of associated encrypted content.

If applicable, the frame of audio data shall be decoded according to the selected coding format.

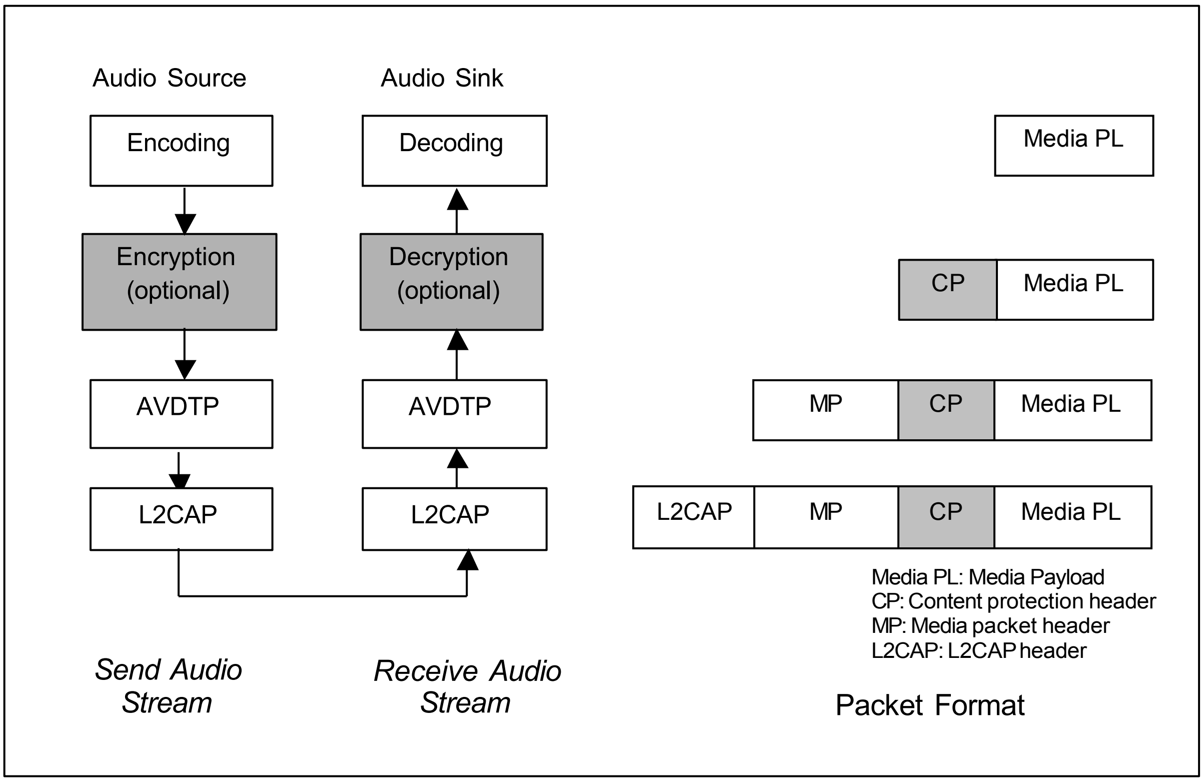


Figure 3.1: Block Diagram of Audio Streaming Procedures and the Packet Format

4 Audio Codec Interoperability Requirements

4.1 Overview

This chapter defines necessary information specific for audio codec. In Section 4.2 definition of codecs used in this profile version 1.0 and their requirements are fully described. Additional information about codec introduced after A2DP version 1.0 is described in Bluetooth Assigned Numbers[8].

Remaining sections provide reference for each codec as well as the following information:

- *Audio codec capabilities* define the capability field and its parameters necessary for signalling procedures in the streaming set up. Related procedures in GAVDP are *Connection Establishment* and *Change Parameters* procedures.
- *Media packet header requirements* define codec specific parameters in the media packet header, which shall be added to the media payload in the AVDTP entity. (See Figure 3.1)
- *Media payload format* defines the codec specific payload format in the AVDTP packet, which shall be used in the *Audio Streaming* procedures in Section 3.2. See also Figure 3.1.

4.2 Support of Codecs

Table 4.1 shows supported *Mandatory* and *Optional* codecs in this profile.

Codec Type	**Support**	**Ref.**
SBC	M	4.3
MPEG-1,2 Audio	O	4.4
MPEG-2,4 AAC	O	4.5
ATRAC family	O	4.6

Table 4.1: Supported codecs

The following codecs are treated as *Non-A2DP* codecs:

- The codecs that are not on Table 4.1.
- The *Mandatory* or *Optional* codecs on Table 4.1 used in non-conforming way.

Requirements for the use of *Non-A2DP* codecs are defined in Section 4.2.3 and 4.7.

4.2.1 Mandatory Codec

The A2DP mandates low complexity subband codec (SBC) to ensure the interoperability. The device shall implement a SBC encoder when the device is the **SRC**, and a SBC decoder when the device is the **SNK**. All other codecs in this document are called *Non-Mandatory* codecs.

4.2.2 Optional codecs

The device may also support *Optional* codecs to maximize its usability. When both **SRC** and **SNK** support the same *Optional* codec, this codec may be used instead of Mandatory codec. *Optional* codecs available in this profile version 1.0 are listed in Table 4.1. To maintain interoperability, the requirements in Section 4.2.4 shall be applied.

4.2.3 Non-A2DP Codecs

The device may support other codecs as *Non-A2DP* codecs. A user of the *Non-A2DP* codec (hereafter the Vendor) oneself defines parameters and any information necessary for use of the codec in A2DP. The profile does not specify anything for *Non-A2DP* codecs, whereas the following requirements are imposed:

1. To maintain interoperability, the requirements in Section 4.2.4 shall be applied.
2. The *Non-A2DP* codec can be upgraded to *Optional* when the following items are prepared;
    - Clear pointer to the specification, test vectors, and related documents
    - Necessary parameters for Signalling

4.2.4 Codec Interoperability Requirements

When the **SRC** wishes to send an audio data whose codec format is not supported by the **SNK**, the data shall be transcoded into SBC. Therefore, the following requirements are applied to the **SRC** when it supports *Non-Mandatory* codecs.

- Transcoding to SBC is only required for any **SRC** input whose format is not supported by the **SNK**

For example, when the **SRC** accepts pre-encoded audio data in the *Non-Mandatory* codec format, the **SRC** shall have a decoder of this *Non-Mandatory* codec as well as a SBC encoder for transcoding.

4.2.5 Audio Codec Type Field Values

Refer to Bluetooth Assigned Numbers[8] for audio codec types available in this profile. Message format of audio codec capabilities is defined in Section 8.19.5 of AVDTP[4].

The following section defines audio codec parameters and formats required for audio streaming on the Bluetooth link.

4.3 SBC

4.3.1 Reference

SBC is mandatory to support in this profile. The SBC specification is a part of the Bluetooth specification. The codec specification is attached in Chapter 12 (Appendix B) of this profile.

4.3.2 Codec Specific Information Elements

Figure 4.1 shows *Codec Specific Information Elements* for SBC used in the signalling procedures. For reference, see Section 8.19.5 of AVDTP[4]. The following section defines the field values and their requirements. The meaning of each value is defined in the SBC specification in Appendix B. If the packet includes improper settings, the error code shall be returned as specified in Section 5.1.3.

7	6	5	4	3	2	1	0	
Sampling Frequency				Channel Mode				Octet0
Block Length				Subbands		Allocation Method		Octet1
Minimum Bitpool Value								Octet2
Maximum Bitpool Value								Octet3

Figure 4.1: Codec Specific Information Elements for SBC

Note: In the Get Capabilities Response of AVDTP, one or more bits may be defined/set in each field. On the other hand, in the Set Configuration Command and the Reconfigure Command of AVDTP, only one bit shall be defined/set in each field.

4.3.2.1 Sampling Frequency

Table 4.2 shows the value of *Sampling Frequency* field for SBC. For the decoder in the **SNK** the sampling frequencies 44.1 kHz and 48 kHz are mandatory to support. The encoder in the **SRC** shall support at least one of the sampling frequencies of 44.1kHz and 48kHz.

Position	**Sampling Frequency (Hz)**	**Support in SRC**	**Support in SNK**
Octet0; b7	16000	O	O
Octet0; b6	32000	O	O
Octet0; b5	44100	C1	M
Octet0; b4	48000	C1	M
C1: At least one of the values shall be supported			

Table 4.2: Sampling Frequency for SBC

4.3.2.2 Channel Mode

Table 4.3 shows the value of *Channel Mode* field for SBC. For the decoder in the **SNK** all features shall be supported. The encoder in the **SRC** shall support at least MONO and one of DUAL CHANNEL, STEREO and JOINT STEREO modes.

Position	**Channel Mode**	**Support in SRC**	**Support in SNK**
Octet0; b3	MONO	M	M
Octet0; b2	DUAL CHANNEL	C1	M
Octet0; b1	STEREO	C1	M
Octet0; b0	JOINT STEREO	C1	M
C1: At least one of the values shall be supported			

Table 4.3: Channel Mode for SBC

4.3.2.3 Block Length

Table 4.4 shows the value of *Block Length* field for SBC. Both encoder in the **SRC** and decoder in the **SNK** shall support all of the parameters.

Position	Block length	Support in SRC	Support in SNK
Octet1; b7	4	M	M
Octet1; b6	8	M	M
Octet1; b5	12	M	M
Octet1; b4	16	M	M

Table 4.4: Block Length for SBC

4.3.2.4 Subbands

Table 4.5 shows the value of *Number of Subbands* field for SBC. For the decoder in the **SNK**, all features shall be supported. The encoder in the **SRC** shall support at least 8 subbands case.

Position	Number of subbands	Support in SRC	Support in SNK
Octet1; b3	4	O	M
Octet1; b2	8	M	M

Table 4.5: Number of Subbands for SBC

4.3.2.5 Allocation Method

Table 4.6 shows the value of *Allocation Method* field for SBC. For the decoder in the **SNK**, all features shall be supported. The encoder in the **SRC** shall support at least the LOUDNESS method.

Position	Allocation method	Support in SRC	Support in SNK
Octet1; b1	SNR	O	M
Octet1; b0	Loudness	M	M

Table 4.6: Allocation Method for SBC

4.3.2.6 Minimum / Maximum Bitpool Value

The device sets the range of SBC bitpool parameters using *Minimum / Maximum Bitpool Value* fields expressed by 8 bit UiMsbf (Unsigned integer, Most significant bit first), ranging from 2 to 250. In the *Get Capabilities* procedure in AVDTP, the *Minimum / Maximum Bitpool Value* fields contain allowed variable range of the bitpool value in the **ACP**, while in the *Stream Configuration* or *Stream Reconfigure* procedure in AVDTP, the fields contain variable range of the bitpool value that the **INT** expects to send/receive.[1] Using the bitpool value and other codec parameters (sampling frequency, channel mode, block length and the number of subbands), the bit rate and frame length of the audio stream is calculated as shown in Section 12.9.

[1] If *Minimum / Maximum Bitpool Value* fields contain the same number, the bitpool value shall be fixed.

The codec information that determines the bit rate is contained in the SBC frame header and repeatedly sent to the **SNK** associated with audio data stream. The **SRC** is capable of changing the bit rate dynamically by changing the bitpool parameter without suspending. The other parameters can be changed during the *Change Parameters* procedure defined in GAVDP.

The decoder of the **SNK** shall support all possible bitpool values that do not result in excess of the maximum bit rate. This profile limits the available maximum bit rate to 320kb/s for mono, and 512kb/s for two-channel modes.

For the encoder of the **SRC**, it is required to support at least one possible bitpool value. However, it is recommended for the encoder to support the following settings shown in Table 4.7.

	Middle Quality				**High Quality**			
SBC encoder settings*	**Mono**		**Joint Stereo**		**Mono**		**Joint Stereo**	
Sampling frequency (kHz)	44.1	48	44.1	48	44.1	48	44.1	48
Bitpool value	19	18	35	33	31	29	53	51
Resulting frame length (bytes)	46	44	83	79	70	66	119	115
Resulting bit rate (kb/s)	127	132	229	237	193	198	328	345
*Other settings: Block length = 16, Allocation method = Loudness, Subbands = 8								

Table 4.7: Recommended sets of SBC parameters in the ***SRC*** *device*

Note again that the frame length shown in this table is variable according to the bitpool value. For the most efficient use of the transport in L2CAP, the frame length may be adjusted when media payload is constructed. For creation of media payload format using SBC frames, see Section 4.3.4.

4.3.3 Media Packet Header Requirements

4.3.3.1 Timestamp (TS)

The clock frequency necessary to create TS shall be set to the sample rate of the encoded audio data.

If a media payload consists of multiple SBC frames, the TS of the media packet header represent the TS of the first SBC frame. The TS of the following SBC frames shall be calculated using the sample rate and the number of samples per frame per channel.

When a SBC frame is fragmented into multiple media packets, all packets that make up a fragmented SBC frame shall use the same TS.

4.3.3.2 Payload Type (PT)

A payload type in the dynamic range shall be chosen.

4.3.3.3 Marker (M) bit

Set to zero.

4.3.3.4 Extension (X) bit

Not used, set to zero.

4.3.4 Media Payload Format

The media payload for SBC shown in Figure 4.2 consists of SBC specific header and SBC frame(s) defined in the SBC specification.

If the configured MTU size for the transport channel is greater or equal to the SBC frame size + the sum of [Media Payload header size, Content Protection header size (if Content Protection is selected), Media Packet header size], then a media payload shall contain an integral number of complete SBC frames (a).

If this is not the case, and provided that the multiplexing service of AVDTP is not selected, the SBC frame shall be fragmented across several media payloads (b). All fragmented packets, except the last one, shall have the same total data packet size. A media payload always starts with an 8-bit header, which is placed before the SBC data. If the multiplexing service of AVDTP is selected, then it is recommended not to fragment the SBC frame across several media payloads, because AVDTP shall fragment the media payloads across several L2CAP packets if necessary.

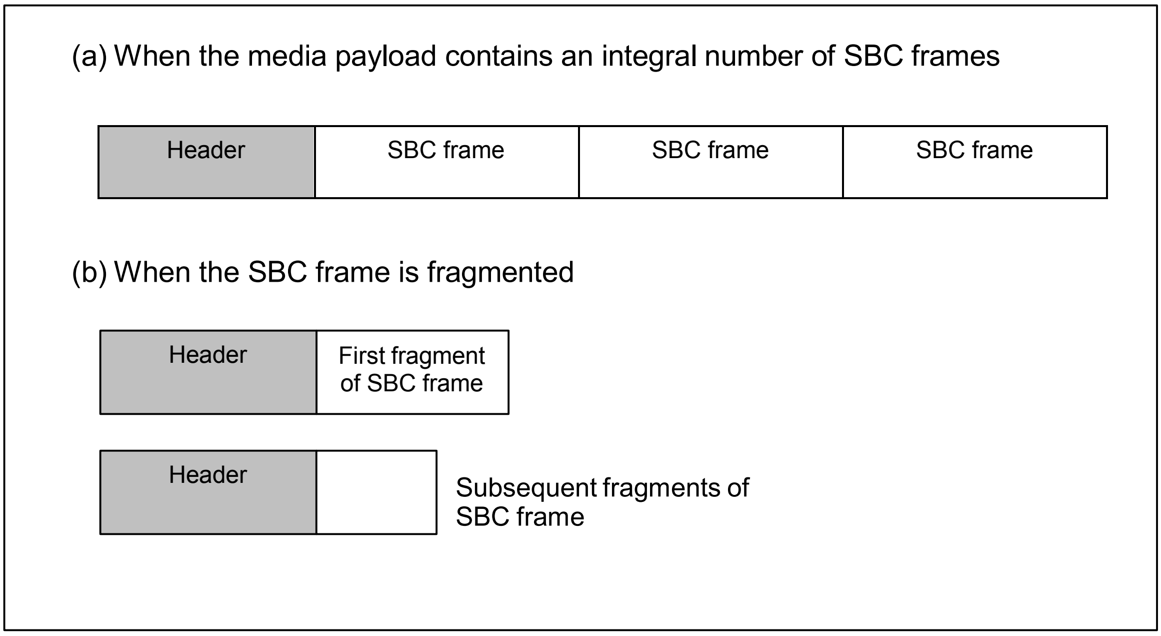

(a) When the media payload contains an integral number of SBC frames
Header
SBC frame
SBC frame
SBC frame
(b) When the SBC frame is fragmented
Header
First fragment of SBC frame
Header
Subsequent fragments of SBC frame

Figure 4.2: Media payload format of SBC

Figure 4.3 shows the media payload header format of SBC.

7	6	5	4	3	2	1	0	
F	S	L	RFA	Number of frames				Octet0

Figure 4.3: Header format of media payload for SBC

- *F bit* – Set to 1 if the SBC frame is fragmented, otherwise set to 0.
- *S bit* – Set to 1 for the starting packet of a fragmented SBC frame, otherwise set to 0.
- *L bit* – Set to 1 for the last packet of a fragmented SBC frame, otherwise set to 0
- *RFA* – See definition Section 1.3.2.1.
- *Number of frames* (4 bits) – If the *F bit* is set to 0, this field indicates the number of frames contained in this packet. If the *F bit* is set to 1, this field indicates the number of remaining fragments, including the current fragment. Thus the last counter value shall be one. For example, if there are three fragments then the counter has value 3,2 and 1 for subsequent fragments. This field is expressed by 4 bit UiMsbf.

4.4 MPEG-1,2 Audio

4.4.1 Reference

For MPEG-1 Audio, refer to [12]. For MPEG-2 Audio, refer to [13].

4.4.2 Codec Specific Information Elements

Figure 4.4 shows *Codec Specific Information Elements* for MPEG-1,2 Audio used in the signalling procedures. For reference, see Section 8.19.5 AVDTP[4]. The following section defines the field values and their requirements. The meaning of each value is defined in [12] and [13]. Support columns in each field value show the requirements to fulfil when this codec is supported. If the packet includes improper settings, the error code shall be returned as specified in Section 5.1.3.

7	6	5	4	3	2	1	0	
Layer			CRC	Channel Mode				Octet0
RFA	MPF	Sampling Frequency						Octet1
VBR	Bit Rate							Octet2
Bit Rate								Octet3

Figure 4.4: Codec Specific Information Elements for MPEG-1,2 Audio

Note: In the Get Capabilities Response of AVDTP, one or more values may be defined/set in each field. On the other hand, in the Set Configuration Command and the Reconfigure Command of AVDTP, only one value shall be defined/set in each field.

4.4.2.1 Layer

Table 4.8 shows the value of *Layer* defined in MPEG-1,2 Audio. The **SRC** and **SNK** shall support at least one of Layer I (mp1), Layer II (mp2) and Layer III (mp3).

Position	Layer	Support in SRC	Support in SNK
Octet0; b7	Layer I (mp1)	C1	C2
Octet0; b6	Layer II (mp2)	C1	C2
Octet0; b5	Layer III (mp3)	C1	C2
C1: At least one of the values shall be supported C2: At least one of the values shall be supported			

Table 4.8: Layers for MPEG-1,2 Audio

4.4.2.2 CRC Protection

Support of *CRC Protection* is mandatory in the **SNK** and optional in the **SRC**.

Position	CRC Protection	Support in SRC	Support in SNK
Octet0; b4	Protection supported	O	M

Table 4.9: CRC Protection assignment for MPEG-1,2 Audio

4.4.2.3 Channel Mode

Table 4.10 shows the value of *Channel Mode* field for MPEG-1,2 Audio. For the decoder in the **SNK** all features shall be supported. The encoder in the **SRC** shall support at least one of MONO*,* DUAL CHANNEL, STEREO and JOINT STEREO modes.

Position	Channel Mode	Support in SRC	Support in SNK
Octet0; b3	MONO	C1	M
Octet0; b2	DUAL CHANNEL	C1	M
Octet0; b1	STEREO	C1	M
Octet0; b0	JOINT STEREO	C1	M
C1: At least one of the values shall be supported			

Table 4.10: Channel Mode for MPEG-1,2 Audio

4.4.2.4 Media Payload Format (MPF)

MPF field indicates the support of media payload format for MPEG-1,2 Audio. It is mandatory to support MPF-1 in Section 4.4.4. The *MPF* field is set to 1 if MPF-2 in Section 4.4.4 is also supported, or if MPF-2 is configured as a transferred media payload format; otherwise it is set to 0.

4.4.2.5 Sampling Frequency

Table 4.11 shows the value of *Sampling Frequency* field for MPEG-1,2 Audio. For the decoder in the **SNK** the sampling frequencies 44.1 kHz and 48 kHz are mandatory to support. The encoder in the **SRC** shall support at least one of the sampling frequencies of 44.1kHz and 48kHz. Other sampling frequencies are optional both for the **SNK** and the **SRC**.

Position	Sampling Frequency (Hz)	Support in SRC	Support in SNK
Octet1; b5	16000	O	O
Octet1; b4	22050	O	O
Octet1; b3	24000	O	O
Octet1; b2	32000	O	O
Octet1; b1	44100	C1	M
Octet1; b0	48000	C1	M
C1: At least one of the values shall be supported			

Table 4.11: Sampling Frequency for MPEG-1,2 Audio

4.4.2.6 VBR

In this profile, support of *VBR* (Variable Bit Rate) for MPEG-1,2 Audio is mandatory for the decoder in the **SNK** and optional for the encoder in the **SRC**. Layer I and Layer II do not mandate this parameter, but most of the actual devices support it commonly.

Position	VBR Support	Support in SRC	Support in SNK
Octet2; b7	VBR supported	O	M

Table 4.12: VBR support for MPEG-1,2 Audio

4.4.2.7 Bit Rate Index

Table 4.13 shows the value of *Bit Rate Index* field for MPEG-1,2 Audio. The index value represents the actual bit rate value defined in the referenced specification. For the decoder in the **SNK** all features shall be supported except for the index value ‘0000’. The encoder in the **SRC** shall support at least one of the index values that are mandatory to support in the **SNK**.

Note that MPEG-1 Layer II (mp2) has restriction in allowed combination of total bit rate and channel mode (for MPEG-1 see Section 2.4.2.3 in [12]). This restriction overrules the support of *Bit Rate Index* shown in Table 4.13.

Position	Bit Rate Index	Support in SRC	Support in SNK
Octet2; b6	‘1110’	C1	M
Octet2; b5	‘1101’	C1	M
Octet2; b4	‘1100’	C1	M
Octet2; b3	‘1011’	C1	M
Octet2; b2	‘1010’	C1	M
Octet2; b1	‘1001’	C1	M
Octet2; b0	‘1000’	C1	M
Octet3; b7	‘0111’	C1	M
Octet3; b6	‘0110’	C1	M
Octet3; b5	‘0101’	C1	M
Octet3; b4	‘0100’	C1	M
Octet3; b3	‘0011’	C1	M
Octet3; b2	‘0010’	C1	M
Octet3; b1	‘0001’	C1	M
Octet3; b0	‘0000’	O	O
C1: At least one of the values shall be supported			

Table 4.13: Bit Rate Index for MPEG-1,2 Audio

4.4.2.8 RFA

See definition Section 1.3.2.1

4.4.3 Media Packet Header Requirements

The media packet header requirements for MPEG-1,2 Audio are contained in the specification of media payload format referenced in Section 4.4.4.

4.4.4 Media Payload Format

MPEG-1,2 Audio uses payload formats defined in [14] and [15]. This profile mandates support of the format in MPF-1. MPF-2 provides more error-robustness for MPEG-1,2 Audio Layer III. See also Section 4.4.2.4. For MPF-1, refer to [14]. For MPF-2, refer to [15].

4.5 MPEG-2, 4 AAC

4.5.1 Reference

For MPEG-2 AAC, refer to [16]. For MPEG-4 AAC, refer to [17].

4.5.2 Codec Specific Information Elements

Figure 4.5 shows *Codec Specific Information Elements* for MPEG-2,4 AAC used in the signalling procedures. For reference, see Section 8.19.5 of AVDTP[4]. The following section defines the field values and their requirements. Support columns in each field value show the requirements to fulfil when this codec is supported. If the packet includes improper settings, the error code shall be returned as specified in Section 5.1.3.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th></th></tr>
<tr><td colspan="8">Object Type</td><td>Octet0</td></tr>
<tr><td colspan="8">Sampling Frequency</td><td>Octet1</td></tr>
<tr><td colspan="4">Sampling Frequency</td><td colspan="2">Channels</td><td colspan="2">RFA</td><td>Octet2</td></tr>
<tr><td>VBR</td><td colspan="7">Bit rate</td><td>Octet3</td></tr>
<tr><td colspan="8">Bit rate</td><td>Octet4</td></tr>
<tr><td colspan="8">Bit rate</td><td>Octet5</td></tr>
</table>

Figure 4.5: Codec Specific Information Elements for MPEG-2,4 AAC

Note: In the Get Capabilities Response of AVDTP, one or more values may be defined/set in each field. On the other hand, in the Set Configuration Command and the Reconfigure Command of AVDTP, only one value shall be defined/set in each field.

4.5.2.1 Object Type

Table 4.14 shows the value of *Object Type* field for MPEG-2,4 AAC. The **SRC** and **SNK** shall support MPEG-2 AAC LC, and other values are optional.

Position	Object Type	Support in SRC	Support in SNK
Octet0; b7	MPEG-2 AAC LC	M	M
Octet0; b6	MPEG-4 AAC LC	O	O
Octet0; b5	MPEG-4 AAC LTP	O	O
Octet0; b4	MPEG-4 AAC scalable	O	O
Octet0; b3	RFA	–	–
Octet0; b2	RFA	–	–
Octet0; b1	RFA	–	–
Octet0; b0	RFA	–	–

Table 4.14: Object Type for MPEG-2,4 AAC

4.5.2.2 Sampling Frequency

Table 4.15 shows the value of *Sampling Frequency* field for MPEG-2,4 AAC. For the decoder in the **SNK** the sampling frequencies 44.1 kHz and 48 kHz are mandatory to support. The encoder in the **SRC** shall support at least one of the sampling frequencies of 44.1kHz and 48kHz. Other sampling frequencies are optional both for **SNK** and **SRC**.

Position	Sampling Frequency (Hz)	Support in SRC	Support in SNK
Octet1; b7	8000	O	O
Octet1; b6	11025	O	O
Octet1; b5	12000	O	O
Octet1; b4	16000	O	O
Octet1; b3	22050	O	O
Octet1; b2	24000	O	O
Octet1; b1	32000	O	O
Octet1; b0	44100	C1	M
Octet2; b7	48000	C1	M
Octet2; b6	64000	O	O
Octet2; b5	88200	O	O
Octet2; b4	96000	O	O
C1: At least one of the values shall be supported			

Table 4.15: Sampling Frequency field for MPEG-2,4 AAC

4.5.2.3 Channels

Table 4.16 shows the value of *Channels* field for MPEG-2,4 AAC. The **SNK** shall support both of channels, while the **SRC** shall support at least one of the channels.

Position	Channels	Support in SRC	Support in SNK
Octet2; b3	1	C1	M
Octet2; b2	2	C1	M
C1: At least one of the values shall be supported			

Table 4.16: Channels field for MPEG-2,4 AAC

4.5.2.4 Bit rate

Bit rate field is assigned for the bit rate in bits per second in case of a constant rate stream, or the maximum peak bit rate (measured per frame) in case of a variable bit rate stream. A value of 0 indicates that the bit rate is not known. The field is expressed as a 23 bit UiMsbf as described in the following figure.

Octet 3 (bits 6... 0)							Octet 4								Octet 5							
b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

Figure 4.6: Bit order for Bit rate field of MPEG-2,4 AAC

4.5.2.5 VBR

Support of *VBR* (Variable Bit Rate) is mandatory for the decoder in the **SNK** and optional for the encoder in the **SRC**.

Position	VBR Support	Support in SRC	Support in SNK
Octet3; b7	VBR supported	O	M

Table 4.17: VBR support for MPEG-2,4 AAC

4.5.2.6 RFA

See definition Section 1.3.2.1.

4.5.3 Media Packet Header Requirements

The media packet header requirements for MPEG-2,4 AAC are contained in the specification of media payload format referenced in Section 4.5.4.

4.5.4 Media Payload Format

MPEG-2,4 AAC uses the media payload format defined in [18]. The specification defines the payload format only for MPEG-4 audio; in use of MPEG-2 AAC LC, the audio stream shall be transformed to MPEG-4 AAC LC in the **SRC** by modifying the codec information and adapted into MPEG-4 LATM format before being put into Media Payload Format. The **SNK** shall retransform the stream into MPEG-2 AAC LC, if necessary.[2] For details, see [16] and [17].

4.6 ATRAC family

4.6.1 Reference

ATRAC family is proprietary codec owned by Sony Corporation. Licensed users obtain the specifications of this codec. For details of license, contact Sony Corporation through the following e-mail address: bt-atrac3@jp.sony.com.

4.6.2 Codec Specific Information Elements

Figure 4.7 shows *Codec Specific Information Elements* for ATRAC family used in the signalling procedures. For reference, see Section 8.19.5 of AVDTP[4]. The following section defines the field values and their requirements. Support columns in each field value show the requirements to fulfil when this codec is supported. If the packet includes improper settings, the error code shall be returned as specified in Section 5.1.3.

[2] When the MPEG-4 AAC LC is supported in the **SNK**, it is possible to decode the data as it is.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th></th></tr>
<tr><td colspan="3">Version</td><td colspan="3">Channel Mode</td><td colspan="2">RFA</td><td>Octet0</td></tr>
<tr><td colspan="2">RFA</td><td colspan="2">Fs</td><td>VBR</td><td colspan="3">Bit Rate</td><td>Octet1</td></tr>
<tr><td colspan="8">Bit Rate</td><td>Octet2</td></tr>
<tr><td colspan="8">Bit Rate</td><td>Octet3</td></tr>
<tr><td colspan="8" rowspan="2">Maximum SUL</td><td>Octet4</td></tr>
<tr><td>Octet5</td></tr>
<tr><td colspan="8">RFA</td><td>Octet6</td></tr>
</table>

Figure 4.7: Codec Specific Information Elements for ATRAC family

Note: In the Get Capabilities Response of AVDTP, one or more values may be defined/set in each field. On the other hand, in the Set Configuration Command and the Reconfigure Command of AVDTP, only one value shall be defined/set in each field.

4.6.2.1 Version

Table 4.18 shows the value of *Version* field for ATRAC family. The *Version* field contains one specific version of ATRAC family. Therefore, if the device supports both ATRAC and ATRAC3, for example, two sets of *Service Capabilities* shall be exchanged.

Bits			
7	**6**	**5**	**Version**
0	0	1	ATRAC
0	1	0	ATRAC2
0	1	1	ATRAC3
Other values			RFD. See definition Section 1.3.2.2.

Table 4.18: Version for ATRAC family

4.6.2.2 Channel Mode

Table 4.19 shows the value of *Channel Mode* field for ATRAC family. The **SRC** and the **SNK** shall support at least one of the values.

Position	**Channel Mode**	**Support in SRC**	**Support in SNK**
Octet0; b4	Single channel	C1	C2
Octet0; b3	Dual channel	C1	C2
Octet0; b2	Joint stereo	C1	C2
C1,C2: At least one of the values shall be supported. For the additional conditions, refer to the specifications of ATRAC family.			

Table 4.19: Channel Mode for ATRAC family

4.6.2.3 Fs (Sampling Frequency)

Table 4.20 shows the value of *Sampling Frequency* field for ATRAC family. The **SRC** and the **SNK** shall support at least one of the values.

Position	Sampling Frequency (Hz)	Support in SRC	Support in SNK
Octet1; b5	44100	C1	C2
Octet1; b4	48000	C1	C2
C1,C2: At least one of the values shall be supported. For the additional conditions, refer to the specifications of ATRAC family.			

Table 4.20: Sampling Frequency for ATRAC family

4.6.2.4 VBR

Support of *VBR* (Variable Bit Rate) for ATRAC family is optional both for the **SRC** and the **SNK**.

Note that when the *VBR* is supported *Bit Rate Index* field in Section 4.6.2.5 shall be neglected since the device can adopt any bit rate under *Maximum SUL* value described in Section 4.6.2.6. When the *VBR* is not applied the *Bit Rate Index* field explicitly indicates supported bit rate, while the *Maximum SUL* field shall be neglected.

Position	VBR Support	Support in SRC	Support in SNK
Octet1; b3	VBR supported	O	O

Table 4.21: VBR support for ATRAC family

4.6.2.5 Bit Rate Index

Table 4.22 shows the value of *Bit Rate Index* field for ATRAC family. The index value represents the actual bit rate value defined in the referenced specification. At least one of the values shall be supported both for the **SRC** and the **SNK**.

Position	Bit Rate Index	Support in SRC	Support in SNK
Octet1; b2	0x0000	C1	C2
Octet1; b1	0x0001	C1	C2
Octet1; b0	0x0002	C1	C2
Octet2; b7	0x0003	C1	C2
Octet2; b6	0x0004	C1	C2
Octet2; b5	0x0005	C1	C2
Octet2; b4	0x0006	C1	C2
Octet2; b3	0x0007	C1	C2
Octet2; b2	0x0008	C1	C2
Octet2; b1	0x0009	C1	C2
Octet2; b0	0x000a	C1	C2
Octet3; b7	0x000b	C1	C2
Octet3; b6	0x000c	C1	C2
Octet3; b5	0x000d	C1	C2
Octet3; b4	0x000e	C1	C2
Octet3; b3	0x000f	C1	C2
Octet3; b2	0x0010	C1	C2
Octet3; b1	0x0011	C1	C2
Octet3; b0	0x0012	C1	C2
C1,C2: At least one of the values shall be supported. For the additional conditions, refer to the specifications of ATRAC family.			

Table 4.22: Bit Rate Index for ATRAC family

4.6.2.6 Maximum SUL

Sound Unit Length (SUL) is one of the parameters that determine bit rate of the audio stream. The *Maximum SUL* field with 16bits UiMsbf contains the maximum value (expressed in Byte) of the SUL that the decoder in the **SNK** supports. The **SRC** shall send audio streaming data whose SUL is equal to or smaller than that of maximum SUL of the decoder in the **SNK**.

The maximum SUL value in the **SNK** shall be notified to the **SRC** during *Get Capabilities* procedure of AVDTP initiated by the **SRC**, or during *Stream Configuration* procedure of AVDTP initiated by the **SNK**.

4.6.2.7 RFA

See definition Section 1.3.2.1.

4.6.3 Media Packet Header Requirements

4.6.3.1 Timestamp (TS)

The clock frequency necessary to create TS shall be set to the sample rate of the encoded audio data.

If a media payload consists of multiple codec frames of ATRAC family, the TS of the media packet header represent the TS of the first codec frame. The TS of the following codec frames shall be calculated using the sample rate and the number of samples per frame per channel.

4.6.3.2 Payload Type (PT)

A payload type in the dynamic range shall be chosen.

4.6.3.3 Marker (M) bit

Set to zero.

4.6.3.4 Extension (X) bit

Not used, set to zero.

4.6.4 Media Payload Format

Licensed users obtain the specification of Media Payload Format for ATRAC family. See Section 4.6.1.

4.7 Non-A2DP Codec

4.7.1 Reference

Definition and treatment of *Non-A2DP* codec is defined in Section 4.2.3.

4.7.2 Codec Specific Information Elements

Figure 4.8 shows *Codec Specific Information Elements* for *Non-A2DP* codec used in the signalling procedures. For reference, see Section 8.19.5 of AVDTP[4]. If the packet includes improper settings, the error code shall be returned as specified in Section 5.1.3.

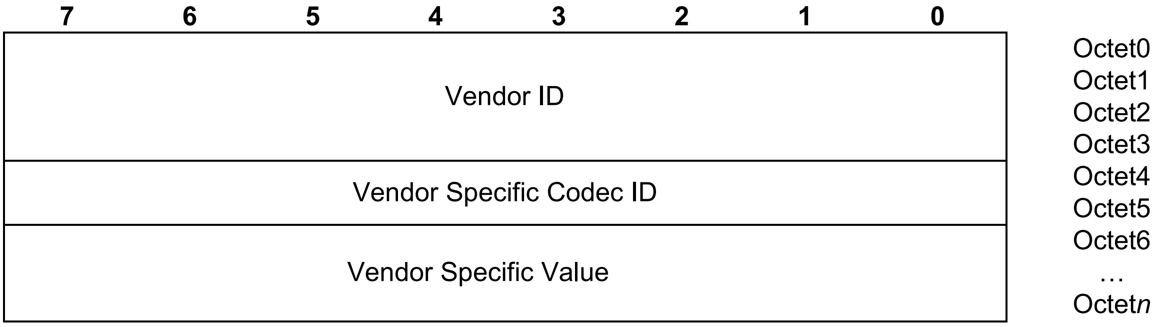


Figure 4.8: Codec Specific Information Elements for Non-A2DP Codec

4.7.2.1 Vendor ID

The 32-bit Vendor ID defined in Bluetooth Assigned Numbers[8] shall be used.

4.7.2.2 Vendor Specific Codec ID

The *Vendor Specific Codec ID* field in Figure 4.8 contains 16-bit codec ID administered by the Vendor.

4.7.2.3 Vendor Specific Value

The *Vendor Specific Value* field in Figure 4.8 contains values specifically defined by the Vendor. Details are out of scope of this profile.

4.7.3 Media Packet Header Requirements

Media Packet Header requirements shall be defined by the Vendor.

4.7.4 Media Payload Format

Media Payload Format shall be defined by the Vendor.

5 GAVDP Interoperability Requirements

This profile requires compliance to the Generic A/V Distribution Profile (GAVDP)[3]. The following text together with the associated sub-clauses defines the requirements with regards to this profile, in addition to the requirements defined in GAVDP.

5.1 AVDTP Interoperability Requirements

5.1.1 Signalling procedures

In the Advanced Audio Distribution Profile, it is mandatory for the **SRC** and optional for the **SNK** to be able to establish a streaming connection, start streaming and release the streaming connection. The **SRC** can assume the role of both **INT** and **ACP**, while the **SNK** device can assume the role of **ACP** and optionally the role of **INT**. Therefore, it is mandatory for **SRC** to support **ACP** role, so that signalling procedures can be manipulated between any combination of a **SRC** device and a **SNK** device.

	Role in GAVDP	Support in SRC	Support in SNK
1	**INT**	M	O
2	**ACP**	M	M

Table 5.1: Roles in GAVDP

5.1.2 Transport Services

Table 5.2 shows support of AVDTP transport capabilities for this profile. In this profile Basic service is mandatory to support.

Item no.	Capability	Ref.	Support
1	Basic service	7.2 in [4]	M
2	Reporting service	7.3 in [4]	O
3	Recovery service	7.4 in [4]	O
4	Multiplexing service	7.5 in [4]	O
5	Robust header compression service	7.6 in [4]	O

Table 5.2: AVDTP transport capabilities

5.1.3 Error Codes

If the *Codec Specific Information Elements* include improper settings, the error code shall be returned as follows. Apart from the error codes specified in GAVDP[3], Table 5.3 below lists additional error codes that shall be used by the application if applicable errors are found in the commands received.

Error ID	Related Signalling command	Related CODEC	Error Abbreviation	Error Description
0xC1	Set Configuration Reconfigure	ALL	INVALID_CODEC_TYPE	Media Codec Type is not valid

0xC2	Set Configuration Reconfigure	ALL	NOT_SUPPORTED_CODEC_TYPE	Media Codec Type is not supported
0xC3	Set Configuration Reconfigure	ALL	INVALID_SAMPLING_FREQUENCY	Sampling Frequency is not valid or multiple values have been selected
0xC4	Set Configuration Reconfigure	ALL	NOT_SUPPORTED_SAMPLING_FREQUENCY	Sampling Frequency is not supported
0xC5	Set Configuration Reconfigure	SBC MPEG-1,2 Audio ATRAC family	INVALID_CHANNEL_MODE	Channel Mode is not valid or multiple values have been selected
0xC6	Set Configuration Reconfigure	SBC MPEG-1,2 Audio ATRAC family	NOT_SUPPORTED_CHANNEL_MODE	Channel Mode is not supported
0xC7	Set Configuration Reconfigure	SBC	INVALID_SUBBANDS	None or multiple values have been selected for Number of Subbands
0xC8	Set Configuration Reconfigure	SBC	NOT_SUPPORTED_SUBBANDS	Number of Subbands is not supported
0xC9	Set Configuration Reconfigure	SBC	INVALID _ALLOCATION_METHOD	None or multiple values have been selected for Allocation Method
0xCA	Set Configuration Reconfigure	SBC	NOT_SUPPORTED_ALLOCATION_METHOD	Allocation Method is not supported
0xCB	Set Configuration Reconfigure	SBC	INVALID_MINIMUM_BITPOOL_VALUE	Minimum Bitpool Value is not valid
0xCC	Set Configuration Reconfigure	SBC	NOT_SUPPORTED_MINIMUM_BITPOOL_VALUE	Minimum Bitpool Value is not supported
0xCD	Set Configuration Reconfigure	SBC	INVALID_MAXIMUM_BITPOOL_VALUE	Maximum Bitpool Value is not valid
0xCE	Set Configuration Reconfigure	SBC	NOT_SUPPORTED_MAXIMUM_BITPOOL_VALUE	Maximum Bitpool Value is not supported
0xCF	Set Configuration Reconfigure	MPEG-1,2 Audio	INVALID_LAYER	None or multiple values have been selected for Layer
0xD0	Set Configuration Reconfigure	MPEG-1,2 Audio	NOT_SUPPORTED_LAYER	Layer is not supported
0xD1	Set Configuration Reconfigure	MPEG-1,2 Audio	NOT_SUPPORTED_CRC	CRC is not supported
0xD2	Set Configuration Reconfigure	MPEG-1,2 Audio	NOT_SUPPORTED_MPF	MPF-2 is not supported
0xD3	Set Configuration Reconfigure	MPEG-1,2 Audio MPEG-2,4 AAC ATRAC family	NOT_SUPPORTED_VBR	VBR is not supported

0xD4	Set Configuration Reconfigure	MPEG-1,2 Audio ATRAC family	INVALID_BIT_RATE	None or multiple values have been selected for Bit Rate
0xD5	Set Configuration Reconfigure	MPEG-1,2 Audio MPEG-2,4 AAC ATRAC family	NOT_SUPPORTED_BIT_RATE	Bit Rate is not supported
0xD6	Set Configuration Reconfigure	MPEG-2,4 AAC	INVALID_OBJECT_TYPE	Either 1) Object type is not valid (b3-b0) or 2) None or multiple values have been selected for Object Type
0xD7	Set Configuration Reconfigure	MPEG-2,4 AAC	NOT_SUPPORTED_OBJECT_TYPE	Object Type is not supported
0xD8	Set Configuration Reconfigure	MPEG-2,4 AAC	INVALID_CHANNELS	None or multiple values have been selected for Channels
0xD9	Set Configuration Reconfigure	MPEG-2,4 AAC	NOT_SUPPORTED_CHANNELS	Channels is not supported
0xDA	Set Configuration Reconfigure	ATRAC family	INVALID_VERSION	Version is not valid
0xDB	Set Configuration Reconfigure	ATRAC family	NOT_SUPPORTED_VERSION	Version is not supported
0xDC	Set Configuration Reconfigure	ATRAC family	NOT_SUPPORTED_MAXIMUM_SUL	Maximum SUL is not acceptable for the Decoder in the SNK.
0xDD	Set Configuration Reconfigure	SBC	INVALID_BLOCK_LENGTH	None or multiple values have been selected for Block Length
0xDE-0xDF				RFD
0xE0	Set Configuration Reconfigure	ALL	INVALID_CP_TYPE	The requested CP Type is not supported.
0xE1	Set Configuration Reconfigure Security Control	ALL	INVALID_CP_FORMAT	The format of Content Protection Service Capability/Content Protection Scheme Dependent Data is not correct.
0xE2-0xFF				RFD

Table 5.3: Error Codes

5.2 L2CAP Interoperability Requirements

For the L2CAP layer, no additions to the requirements as stated in the GAVDP shall apply except for the following requirements.

5.2.1 Maximum Transmission Unit

The minimum MTU that a L2CAP implementation for this profile shall support is 335bytes.[3]

5.2.2 Flush Timeout

Application shall set the appropriate value for responding time to the flush timeout. A small finite value should be used to allow sufficient real-time throughput on the interface.

Remark: Flush timeout can be constrained by the ACL channels when the other profile(s) coexist with A2DP.

5.3 SDP Interoperability Requirements

This profile defines the following service records for the **SRC** and the **SNK** respectively.

The codes assigned to the mnemonics used in the Value column as well as the codes assigned to the attribute identifiers (if not specifically mentioned in the AttrID column) can be found in Bluetooth Assigned Numbers[8].

[3] DH5 packet size equals 339bytes including 4-byte L2CAP header.

Item	Definition	Type	Value	AttrID	Status	Default
Service Class ID List					M	
Service Class #0		UUID	Audio Source		M	
Protocol Descriptor List					M	
Protocol #0		UUID	L2CAP		M	
Parameter #0 for Protocol #0	PSM	Uint 16	PSM= AVDTP		M	
Protocol #1		UUID	AVDTP		M	
Parameter #0 for Protocol #1	Version	Uint 16	0x0100*		M	
Bluetooth Profile Descriptor List					M	
Profile #0		UUID	Advanced Audio Distribution		M	
Parameter #0 for Profile #0	Version	Uint 16	0x0100*[1]		M	
Supported Features	A2DP features flags	Uint 16	Bit 0 = Player Bit 1 = Microphone Bit 2 = Tuner Bit 3 = Mixer Bit 4-15 = RFA The bits for supported features are set to 1. Others are set to 0.		O	
Provider Name	Displayable Text Name	String	Provider Name		O	
Service Name	Displayable Text Name	String	Service-provider defined		O	

*1 Indicating Version 1.0.

Figure 5.1: Service Record for Source

Item	Definition	Type	Value	AttrID	Status	Default
Service Class ID List					M	
Service Class #0		UUID	Audio Sink		M	
Protocol Descriptor List					M	
Protocol #0		UUID	L2CAP		M	
Parameter #0 for Protocol #0	PSM	Uint 16	PSM= AVDTP		M	
Protocol #1		UUID	AVDTP		M	
Parameter #0 for Protocol #1	Version	Uint 16	0x0100*		M	
Bluetooth Profile Descriptor List					M	
Profile #0		UUID	Advanced Audio Distribution		M	
Parameter #0 for Profile #0	Version	Uint 16	0x0100*[1]		M	
Supported Features	A2DP features flags	Uint 16	Bit 0 = Headphone Bit 1 = Speaker Bit 2 = Recorder Bit 3 = Amplifier Bit 4-15 = RFA The bits for supported features are set to 1. Others are set to 0.		O	
Provider Name	Displayable Text Name	String	Provider Name		O	
Service Name	Displayable Text Name	String	Service-provider defined		O	

*1 Indicating Version 1.0.

Figure 5.2: Service Record for Sink

5.4 Link Manager Interoperability Requirements

For the LMP layer, no additions to the requirements as stated in the GAVDP shall apply.

5.5 Link Controller Interoperability Requirements

For the Link Controller layer, the requirements as stated in the GAVDP shall apply. Further more the following packets shall be supported in both **SNK** and **SRC**: DH3, DM3, DH5 and DM5

Note: Requirements described in GAVDP is described for **INT**/**ACP**. For **SRC**, it is mandatory to support both **INT** and **ACP**. For **SNK**, it is mandatory to support **ACP** and it is optional to support **INT**.

5.5.1 Class of Device

The Class of Device field shall be set to the following:

1. Mandatory to set the 'Rendering' bit for the **SNK** and the 'Capturing' bit for the **SRC** in the Service Class field.
2. Recommended to set 'Audio/Video' as Major Device class both for the **SNK** and the **SRC**.
3. Select the appropriate Minor Device class as defined in the Bluetooth Assigned Numbers[8].

6 Generic Access Profile Interoperability Requirements

The Advanced Audio Distribution profile requires compliance to the Generic Access Profile.

There is no change to the requirements as stated in the General Audio/Video Distribution Profile.

Note: Requirements described in GAVDP is described for **INT**/**ACP**. For **SRC**, it is mandatory to support both **INT** and **ACP**. For **SNK**, it is mandatory to support **ACP** and it is optional to support **INT**.

7 Testing

The Advanced Audio Distribution profile requires interoperability test. The details of the test strategy are described in [9]. Tested functionality is defined in [10].

8 References

[1] Bluetooth SIG, Specification of the Bluetooth System, Core, Version 1.1, Part B (Baseband)

[2] Bluetooth SIG, Specification of the Bluetooth System, Profiles, version 1.0, Audio/Video Remote Control Profile

[3] Bluetooth SIG, Specification of the Bluetooth System, Profiles, version 1.0, Generic Audio/Video Distribution Profile

[4] Bluetooth SIG, Specification of the Bluetooth System, Core, version 1.0, Audio/Video Distribution Transport Protocol

[5] Bluetooth SIG, Specification of the Bluetooth System, Core, version 1.1, Part C (LMP)

[6] Bluetooth SIG, Specification of the Bluetooth System, Core, version 1.1, Part D (L2CAP)

[7] Bluetooth SIG, Specification of the Bluetooth System, Core, version 1.1, Part E (SDP)

[8] Bluetooth SIG, Bluetooth Assigned Numbers, http://www.bluetooth.org/assigned-numbers.htm

[9] Bluetooth SIG, Specification of the Bluetooth System, TSS, version 1.0, Test Suite Structure (TSS) and Test Procedures (TP) for Advanced Audio Distribution Profile

[10] Bluetooth SIG, Specification of the Bluetooth System, ICS, version 1.0, Profile ICS proforma for Advanced Audio Distribution Profile

[11] F. de Bont, M. Groenewegen and W. Oomen, “A High Quality Audio-Coding System at 128 kb/s”, 98th AES Convention, Febr. 25 – 28, 1995.

[12] ISO/IEC 11172-3:1993: Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s - Part 3: Audio.

[13] ISO/IEC 13818-3:1998: Information technology – Generic coding of moving pictures and audio – Part 3: Audio

[14] IETF RFC 2250: "RTP Payload Format for MPEG1/MPEG2 Video", http://www.ietf.org/

[15] IETF RFC 3119: "A More Loss-Tolerant RTP Payload Format for MP3 Audio", http://www.ietf.org/

[16] ISO/IEC 13818-7:1997: Information technology – Generic coding of moving pictures and associated information – Part 7: Advanced Audio Coding

[17] ISO/IEC 14496-3:2001: Information technology – Coding of audio-visual objects – Part 3: Audio

[18] IETF RFC 3016: "RTP Payload Format for MPEG-4 Audio/Visual streams", http://www.ietf.org/

[19] Bluetooth SIG, Conformance Test Bitstreams and Reference Implementation of SBC,

http://www.bluetooth.org

9 List of Figures

10 List of Tables

11 Appendix A (Informative): Audio Streaming with Content Protection

This profile does not specify a particular content protection method rather it only provides support for various content protection methods. Specifically, AVDTP provides for the identification and negotiation of a particular content protection method via the *Get Capabilities* and *Stream Configuration* procedures.

The *Security Control* procedure in AVDTP provides for the exchange of the activated content protection method.

12 Appendix B: Technical Specification of SBC

12.1 Introduction

This appendix describes the technical specification of Low Complexity Subband Coding (SBC). SBC is an audio coding system specially designed for Bluetooth AV applications to obtain high quality audio at medium bit rates, and having a low computational complexity. SBC uses 4 or 8 subbands, an adaptive bit allocation algorithm, and simple adaptive block PCM quantizers. The SBC audio coding system is based on an earlier system, which was presented on [11]. Reference executables of both the encoder and the decoder of SBC codec are available in [19] for Win32 platform. For further information, refer to Section 9.4.3 in [9]. Note that source code of SBC neither encoder or decoder is available as part of the specification.

12.2 Glossary

Term	Description
frame	Basic unit that can be decoded independently
bit_count	A bit counter that keeps track of the number of bits
bitneed	A counter that represents the remaining bits during the bit allocation process

Table 12.1: Glossary

12.3 Symbols and Abbreviations

12.3.1 Arithmetic Operators

Operator	Description
+	Addition
-	Subtraction (as a binary operator) or negation (as a unary operator)
++	Increment
--	Decrement
*	Multiplication
^	Power
/	Division
div	Integer division
mod	Modulo operation
$\lfloor x \rfloor$	Round x towards minus infinity
$\lceil x \rceil$	Round x towards plus infinity
sin(x)	Sine of x
cos(x)	Cosine of x
exp(x)	Exponential e^x
pow(x,y)	Exponential x^y
$\sqrt{x}$	Square root of x

Table 12.2: Arithmetic operators

12.3.2 Logical Operators

Operator	**Description**
||	Logical OR
&&	Logical AND
!	Logical NOT

Table 12.3: Logical operators

12.3.3 Relation Operators

Operator	**Description**
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
==	Equal to
!=	Not equal to
max()	The maximum in the argument list
min()	The minimum in the argument list
x?y:z	If x is true then y else z

Table 12.4: Relation operators

12.3.4 Bitwise Operators

Operator	**Description**
&	AND
|	OR
>>	Shift right with sign extension
<<	Shift left with zero fill

Table 12.5: Bitwise operators

12.3.5 Assignment

Operator	**Description**
=	Assignment operator

Table 12.6: Assignment

12.3.6 Mnemonics

The following mnemonics are defined to describe the different data types used in the coded bit-stream.

Mnemonic	**Description**
Char8	Character of 8 bits
UiMsbf	Unsigned integer, Most significant bit first
SiMsbf	Signed integer, Most significant bit first
BsMsbf	Bit-stream, Most significant bit first
PCM	Pulse Code Modulation
na	Not available

Table 12.7: Mnemonics

12.3.7 Constants

Constant	Description
π	3.14159265358...

Table 12.8: Constants

12.3.8 Ranges

Range	Description
[0, 10]	A number in the range of 0 up to and including 10
[0, 10>	A number in the range of 0 up to but excluding 10

Table 12.9: Ranges

12.3.9 Number Notation

Number notation	Description
%X	Binary number representation (e.g. %01111100)
$X	Hexadecimal number representation (e.g. $7C)
X	Numbers with no prefix use decimal representation (e.g. 124.43 or 1.2443E+02)

Table 12.10: Number notation

12.4 Syntax

Syntax	No. of bits	Mnemonic

audio_frame()
{
    frame_header()
    scale_factors()
    audio_samples()
    padding()
}		

Table 12.11: Syntax of audio_frame

Syntax	No. of bits	Mnemonic
frame_header()		
{		
syncword	8	BsMsbf
sampling_frequency	2	UiMsbf
blocks	2	UiMsbf
channel_mode	2	UiMsbf
allocation_method	1	UiMsbf
subbands	1	UiMsbf
bitpool	8	UiMsbf
crc_check	8	UiMsbf
If (channel_mode==JOINT_STEREO)		
{		
for (sb=0;sb<nrof_subbands-1;sb++)		
{		
join[sb]	1	UiMsbf
}		
RFA	1	UiMsbf
}		
}		

Table 12.12: Syntax of frame_header

Syntax	No. of bits	Mnemonic
scale_factors()		
{		
for (ch=0;ch<nrof_channels;ch++)		
{		
for (sb=0;sb<nrof_subbands;sb++)		
{		
scale_factor[ch][sb]	4	UiMsbf
}		
}		
}		

Table 12.13: Syntax of scale_factors

Syntax	No. of bits	Mnemonic
audio_samples()		
{		
for (blk=0;blk<nrof_blocks;blk++)		
{		
for (ch=0;ch<nrof_channels;ch++)		
{		
for (sb=0;sb<nrof_subbands;sb++)		
{		
if (bits[ch][sb]!=0)		
{		
audio_sample[blk][ch][sb]	1..16	UiMsbf
}		
}		
}		
}		
}		

Table 12.14: Syntax of audio_samples

Syntax	No. of bits	Mnemonic

padding()
{
    while ((bit_count mod 8)!=0)
    {
        padding_bit
        bit_count++
    }
}
	
	1
		
		UiMsbf

Table 12.15: Syntax of padding

12.5 Semantics

12.5.1 Frame_header

syncword -- The 8 bit string %10011100 or $9C.

sampling_frequency -- Two bits to indicate the sampling frequency with which the stream has been encoded. The sampling frequency f_s is selected conforming to the table below.

sampling_frequency	**f_s (kHz)**
00	16
01	32
10	44.1
11	48

Table 12.16: sampling_frequency

blocks -- Two bits to indicate the block size with which the stream has been encoded. The block size nrof_blocks is selected conforming to the table below.

blocks	**nrof_blocks**
00	4
01	8
10	12
11	16

Table 12.17: blocks

channel_mode -- Two bits to indicate the channel mode that has been encoded. The variable nrof_channels is derived from this information.

channel_mode	**channel mode**	**nrof_channels**
00	MONO	1
01	DUAL_CHANNEL	2
10	STEREO	2
11	JOINT_STEREO	2

Table 12.18: channel_mode

allocation_method -- One bit to indicate the bit allocation method.

Allocation_method	bit allocation method
0	LOUDNESS
1	SNR

Table 12.19: allocation_method

subbands -- One bit to indicate the number of subbands with which the stream has been encoded. The variable nrof_subbands is derived from this information.

Subbands	nrof_subbands
0	4
1	8

Table 12.20: subbands

bitpool -- This is a 8 bit integer to indicate the size of the bit allocation pool that has been used for encoding the stream. The value of the **bitpool** field shall not exceed 16 * **nrof_subbands** for the MONO and DUAL_CHANNEL channel modes and 32 * **nrof_subbands** for the STEREO and JOINT_STEREO channel modes.

crc_check – This 8 bits parity-check word is used for error detection within the encoded stream.

RFA – See definition Section 1.3.2.1.

join[sb] -- One bit to indicate whether joint stereo has been used in subband sb. Equals %1 if the subband has been encoded in joint stereo, and equals %0 if the subband has been encoded in stereo. Join[nrof_subbands-1] always equals %0.

12.5.2 scale_factors

scale_factor[ch][sb] -- Four bits containing the scale factor with which the samples of channel ch in subband sb shall be multiplied.

12.5.3 audio_samples

audio_samples[blk][ch][sb] -- These bits represent the audio sample of block blk in channel ch for subband sb.

12.5.4 padding

padding_bit -- Bits of value %0 that are used to pad the length of an audio_frame to an integral number of Bytes.

12.6 Decoding Processes

In Figure 12.1 the operation of the decoder is illustrated. On the basis of the scale factors the bit allocation is calculated. For the MONO and DUAL_CHANNEL the bit allocation is calculated for each channel independantly (see Section 12.6.3.1). For the STEREO and JOINT_STEREO channel modes the allocation calculation for the two channels is combined (see Section 12.6.3.2). Then the number of quantization levels are derived for each subband, the subband samples are calculated and finally, via a polyphase synthesis filter, the PCM output is generated. This process is further described in this section.

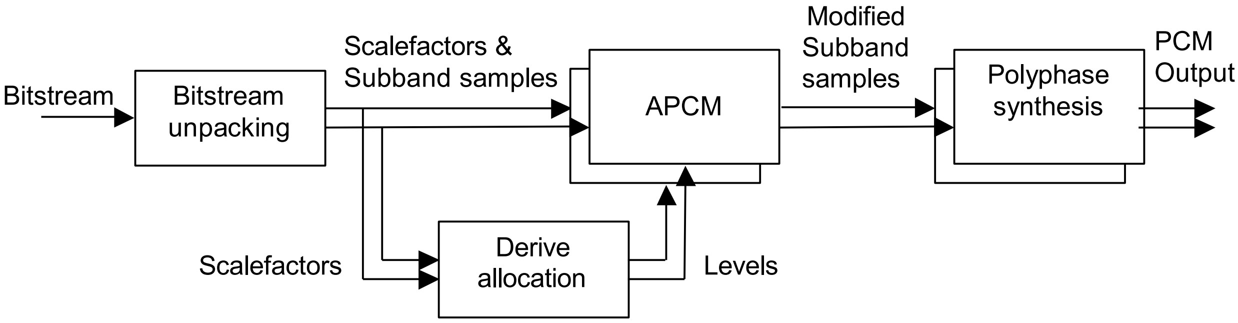


Figure 12.1: Diagram of the decoder

12.6.1 Frame Header

The frame_header contains the configuration with which the bit-stream has been encoded.

12.6.1.1 CRC Check

To detect transmission errors, a CRC check is performed. All the bits of the frame_header, except for the syncword and the crc_check, plus all the bits of the scale_factors are included. The error detection method used is “CRC-8” with generator polynomial

$$G(X) = X^8 + X^4 + X^3 + X^2 + 1 \qquad \text{(CRC-8).}$$

The CRC method is depicted in the CRC-check diagram given in Figure 12.2. The initial state of the shift register is $0F. All bits included in the CRC check are input to the circuit shown in the figure. After each bit is input, the shift register is shifted by one bit. After the last shift operation, the outputs $b_{n-1}\ldots b_0$ constitute a word to be compared with the CRC-check word in the stream. If the words are not identical, a transmission error has occurred in the fields on which the CRC check has been applied. To avoid annoying distortions, application of a concealment technique, such as muting of the actual frame or repetition of the previous frame is recommended.

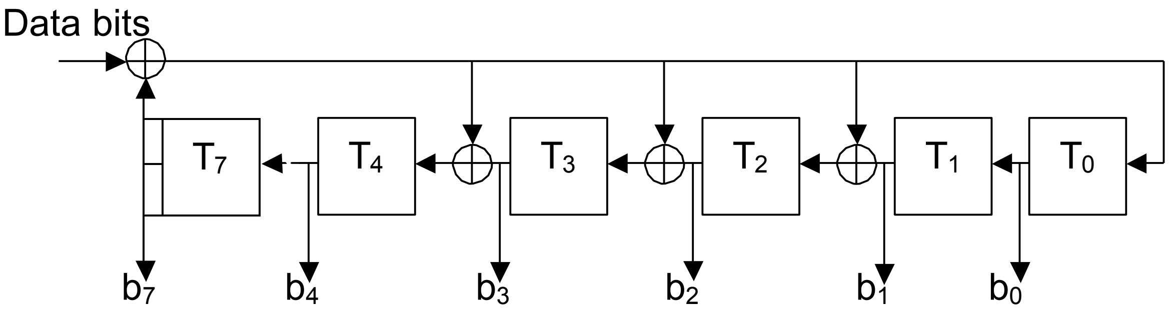


Figure 12.2: CRC-check diagram. The addition blocks represent "exclusive or" gates.

12.6.2 Scale Factors

The actual scaling factor for channel ch and subband sb is calculated according to

$$scalefactor[ch][sb] = pow(2.0, (scale\_factor[ch][sb] + 1)).$$

12.6.3 Bit Allocation

12.6.3.1 Mono and Dual_Channel Bit Allocation

For these two channel modes the bit allocation is calculated for each channel independently and is derived from the scale factors.

In the first step, bitneed values are derived from the scale factors according to the following pseudo code for each channel independently. The tables offset4 and offset8 are in Section 12.8.

```
if (allocation_method==SNR)
{
    for (sb=0;sb<nrof_subbands;sb++)
    {
        bitneed[ch][sb] = scale_factor[ch][sb];
    }
}
else
{
    for (sb=0;sb<nrof_subbands;sb++)
    {
        if (scale_factor[ch][sb] == 0)
        {
            bitneedn[ch][sb] = -5;
        }
        else
        {
            if (nrof_subbands == 4)
            {
                loudness = scale_factor[ch][sb] – offset4[sampling_frequency][sb];
            }
            else
            {
                loudness = scale_factor[ch][sb] – offset8[sampling_frequency][sb];
            }
            if (loudness > 0)
            {
                bitneed[ch][sb] = loudness div 2;
            }
            else
            {
                bitneed[ch][sb] = loudness;
            }
        }
    }
}
```

Then the maximum bitneed index is searched for

```
max_bitneed=0;
for (sb=0;sb<nrof_subbands;sb++)
    if (bitneed[ch][sb] > max_bitneed)
        max_bitneed=bitneed[ch][sb];
```

Next an iterative process finds out how many bitslices fit into the bitpool

```
bitcount=0;
slicecount=0;
bitslice=max_bitneed+1; /* init just above the largest sf */
do{
    bitslice--;
    bitcount+=slicecount;
    slicecount=0;
    for (sb=0;sb<nrof_subbands;sb++)
    {
        if((bitneed[ch][sb]>bitslice+1)&&(bitneed[ch][sb]<bitslice+16))
            slicecount++;
        else if(bitneed[ch][sb]==bitslice+1)
            slicecount+=2;
    }
} while (bitcount+slicecount<bitpool);
if (bitcount+slicecount==bitpool)
{
    bitcount+=slicecount;
    bitslice--;
}
```

Thereafter, bits are distributed until the last bitslice is reached

```
for (sb=0;sb<nrof_subbands;sb++)
    if(bitneed[ch][sb]<bitslice+2)
        bits[ch][sb]=0;
    else
        bits[ch][sb]=min(bitneed[ch][sb]-bitslice,16);
```

The remaining bits are allocated starting at subband 0

```
sb=0;
while(bitcount < bitpool && sb < nrof_subbands)
{
      if((bits[ch][sb]>=2)&&(bits[ch][sb]<16))
      {
            bits[ch][sb]++;
            bitcount++;
      }
      else if((bitneed[ch][sb]==bitslice+1)&&(bitpool>bitcount+1))
      {
            bits[ch][sb]=2;
            bitcount+=2;
      }
      sb++;
}
sb=0;
while(bitcount < bitpool && sb < nrof_subbands)
{
   if (bits[ch][sb] < 16)
   {
      bits[ch][sb]++;
      bitcount++;
   }
   sb++;
 }
```

12.6.3.2 Stereo and Joint_Stereo Bit Allocation

For these two channel modes the bit allocation calculation for the two channels is combined and it is derived from the scale factors of the both channels.

In the first step bitneed values are calculated from the scale factors according to the following pseudo code. The tables offset4 and offset8 are in Section 12.8.

```
if (allocation_method==SNR)
{
    for (ch=0;ch<2;ch++)
    {
        for (sb=0;sb<nrof_subbands;sb++)
        {
            bitneed[ch][sb] = scale_factor[ch][sb];
        }
    }
}
else
{
    for (ch=0;ch<2;ch++)
    {
        for (sb=0;sb<nrof_subbands;sb++)
        {
            if (scale_factor[ch][sb] == 0)
            {
                bitneed[ch][sb] = -5;
            }
            else
            {
                if (nrof_subbands == 4)
                {
                    loudness = scale_factor[ch][sb] – offset4[sampling_frequency][sb];
                }
                else
                {
                    loudness = scale_factor[ch][sb] – offset8[sampling_frequency][sb];
                }
                if (loudness > 0)
                {
                    bitneed[ch][sb] = loudness div 2;
                }
                else
                {
                    bitneed[ch][sb] = loudness;
                }
            }
        }
    }
}
```

Then the maximum bitneed index is searched for

```
max_bitneed=0;
    for (ch=0;ch<2;ch++)
        for (sb=0;sb<nrof_subbands;sb++)
            if (bitneed[ch][sb] > max_bitneed)
                max_bitneed=bitneed[ch][sb];
```

Next an iterative process finds out how many bitslices fit into the bitpool

```
bitcount=0;
slicecount=0;
bitslice=max_bitneed+1; /* init just above the largest sf */
do{
    bitslice--;
    bitcount+=slicecount;
    slicecount=0;
    for (ch=0;ch<2;ch++)
        for (sb=0;sb<nrof_subbands;sb++)
            if((bitneed[ch][sb]>bitslice+1)&&(bitneed[ch][sb]<bitslice+16))
                slicecount++;
            else if(bitneed[ch][sb]==bitslice+1)
                slicecount+=2;
} while (bitcount+slicecount<bitpool);
if (bitcount+slicecount==bitpool)
{
    bitcount+=slicecount;
    bitslice--;
}
```

Thereafter bits are distributed until the last bitslice is reached

```
for (ch=0;ch<2;ch++)
{
    for (sb=0;sb<nrof_subbands;sb++)
    {
        if(bitneed[ch][sb]<bitslice+2)
        {
            bits[ch][sb]=0;
        }
        else
        {
            bits[ch][sb]=min(bitneed[ch][sb]-bitslice,16);
        }
    }
}
```

The remaining bits are allocated starting with subband 0 of the first channel.

```
ch=0;sb=0;
while(bitcount < bitpool && sb < nrof_subbands)
{
      if((bits[ch][sb]>=2)&&(bits[ch][sb]<16))
      {
            bits[ch][sb]++;
            bitcount++;
      }
      else if((bitneed[ch][sb]==bitslice+1)&&(bitpool>bitcount+1))
      {
            bits[ch][sb]=2;
            bitcount+=2;
      }
      if (ch == 1)
      {
            ch = 0;
            sb++;
      }
      else
      {
            ch = 1;
      }
}
ch=0;sb=0;
while(bitcount < bitpool && sb < nrof_subbands)
{
    if (bits[ch][sb] < 16)
    {
        bits[ch][sb]++;
        bitcount++;
    }
    if (ch == 1)
    {
        ch = 0;
        sb++;
    }
    else
    {
        ch = 1;
    }
 }
```

12.6.4 Reconstruction of the Subband Samples

```
for (ch=0;ch<nrof_channels;ch++)
    for (sb=0;sb<nrof_subbands;sb++)
        levels[ch][sb] = pow(2.0,bits[ch][sb])-1;

for (blk=0;blk< nrof_blocks;blk++)
{
    for (ch=0;ch<nrof_channels;ch++)
    {
        for (sb=0;sb<nrof_subbands;sb++)
        {
            if (levels[ch][sb] > 0)
            {
                sb_sample[blk][ch][sb] = scalefactor[ch][sb] * ((audio_sample[blk][ch][sb]*2.0+1.0) /
                                    levels[ch][sb]-1.0);
            }
            else
            {
                sb_sample [blk][ch][sb]=0;
            }
        }
    }
}
```

12.6.5 Joint Processing

For the JOINT_STEREO channel mode, the subbands that are transmitted in joint stereo mode shall be calculated according to:

```
for (blk=0;blk< nrof_blocks;blk++)
{
    for (sb=0;sb<nrof_subbands;sb++)
    {
        if ((channel_mode==JOINT_STEREO) && (join[sb]==1))
        {
            sb_sample[blk][0][sb] = sb_sample[blk][0][sb] + sb_sample[blk][1][sb];
            sb_sample[blk][1][sb] = sb_sample[blk][0][sb] – 2 * sb_sample[blk][1][sb];
        }
    }
}
```

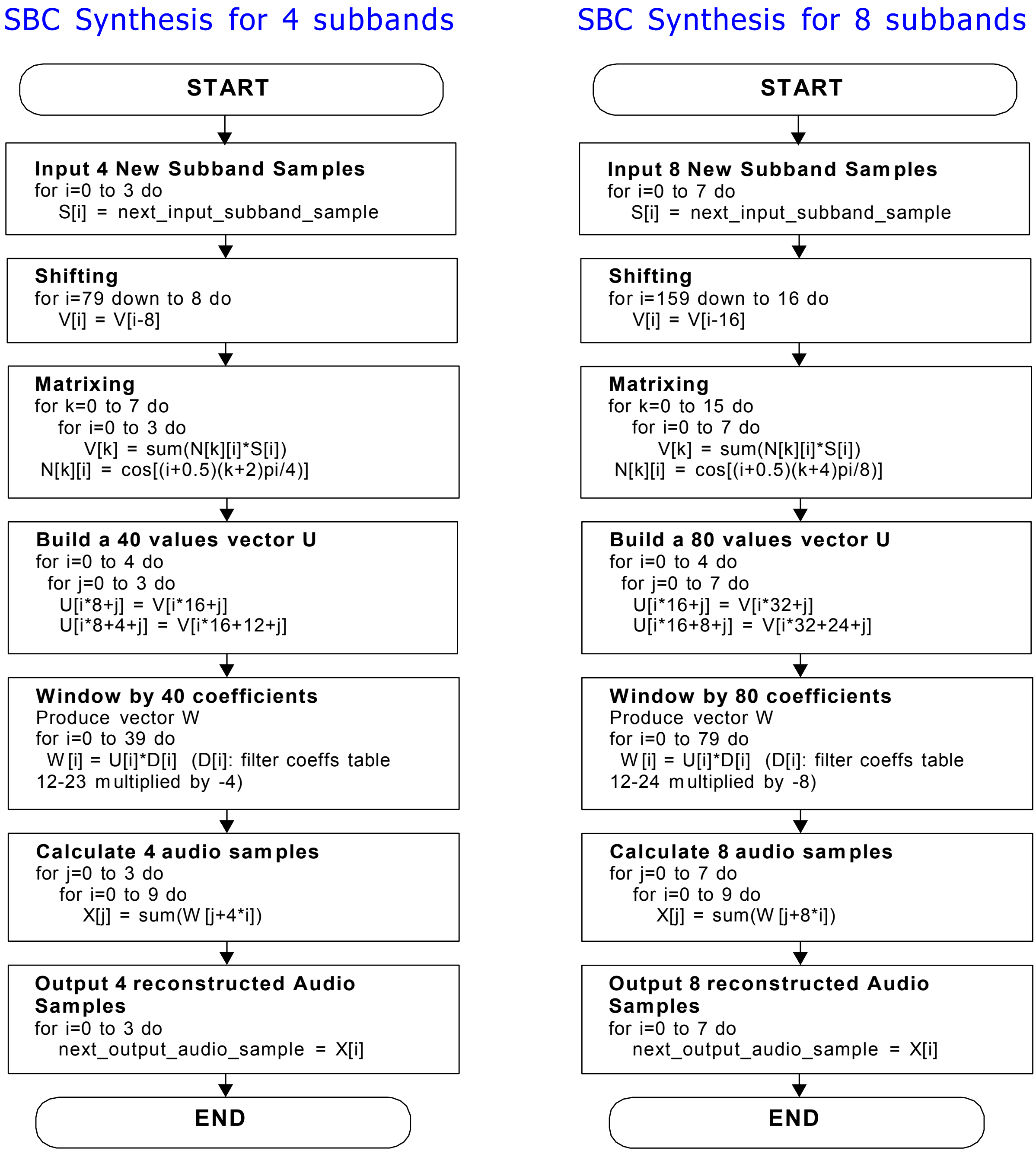
12.6.6 Synthesis Filter

Synthesis of the decoded output is calculated for each channel separately. For each block of decoded subband samples the synthesis filter shall be applied to calculate nrof_subbands consecutive audio samples. The synthesis filter is a polyphase filterbank according to

$$h_m[n] = h_p[n]\cos\left(\left(m + \frac{1}{2}\right)\cdot\left(n + \frac{M}{2}\right)\cdot\frac{\pi}{M}\right), \quad m = [0, M-1], n = [0, L-1],$$

with M=nrof_subbands and L=10*nrof_subbands. The prototype filters (h_p) for both M=4 and M=8 are in Section 12.8. This synthesis filterbank has the same structure as the one that is used in [12].

For more details the reader is refered to Section 2.4.3.2.2, "Synthesis subband filter" in [12]. The sb_sample[blk][ch][sb] values, as calculated in Section 12.6.4, "Reconstruction of the Subband Samples" and Section 12.6.5, "Joint Processing", are the input of the synthesis filter. The output of the synthesis filter are the decoded audio output samples. A detailed filter block diagram can be found in Figure 12.3.



```
Input 4 New Subband Samples
for i=0 to 3 do
   S[i] = next_input_subband_sample

Shifting
for i=79 down to 8 do
   V[i] = V[i-8]

Matrixing
for k=0 to 7 do
   for i=0 to 3 do
      V[k] = sum(N[k][i]*S[i])
 N[k][i] = cos[(i+0.5)(k+2)pi/4)]

Build a 40 values vector U
for i=0 to 4 do
 for j=0 to 3 do
   U[i*8+j] = V[i*16+j]
   U[i*8+4+j] = V[i*16+12+j]

Window by 40 coefficients
Produce vector W
for i=0 to 39 do
 W[i] = U[i]*D[i]  (D[i]: filter coeffs table
12-23 multiplied by -4)

Calculate 4 audio samples
for j=0 to 3 do
   for i=0 to 9 do
      X[j] = sum(W[j+4*i])

Output 4 reconstructed Audio
Samples
for i=0 to 3 do
   next_output_audio_sample = X[i]
```

```
Input 8 New Subband Samples
for i=0 to 7 do
   S[i] = next_input_subband_sample

Shifting
for i=159 down to 16 do
   V[i] = V[i-16]

Matrixing
for k=0 to 15 do
   for i=0 to 7 do
      V[k] = sum(N[k][i]*S[i])
 N[k][i] = cos[(i+0.5)(k+4)pi/8)]

Build a 80 values vector U
for i=0 to 4 do
 for j=0 to 7 do
   U[i*16+j] = V[i*32+j]
   U[i*16+8+j] = V[i*32+24+j]

Window by 80 coefficients
Produce vector W
for i=0 to 79 do
 W[i] = U[i]*D[i]  (D[i]: filter coeffs table
12-24 multiplied by -8)

Calculate 8 audio samples
for j=0 to 7 do
   for i=0 to 9 do
      X[j] = sum(W[j+8*i])

Output 8 reconstructed Audio
Samples
for i=0 to 7 do
   next_output_audio_sample = X[i]
```

These Flow Diagrams are adapted from Figure A.2 and paragraph 2.4.3.2.2 in ISO/IEC 11172-3

Figure 12.3: Flow Diagrams of the Synthesis Filter

12.7 Encoding Processes

In Figure 12.4 the operation of the encoder is illustrated. Via a polyphase analysis filter the input PCM is split into subband signals. For each subband a scale factor is calculated. On the basis of the scale factors the bit allocation, and from there the levels are derived for each subband. Then the subband samples are scaled and quantized and finally, a bitstream is generated. This process is further described in this section.

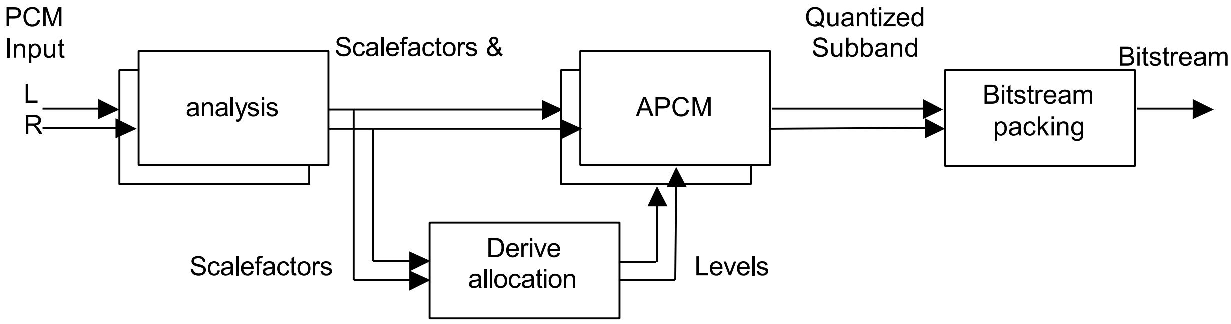


Figure 12.4: Diagram of the encoder

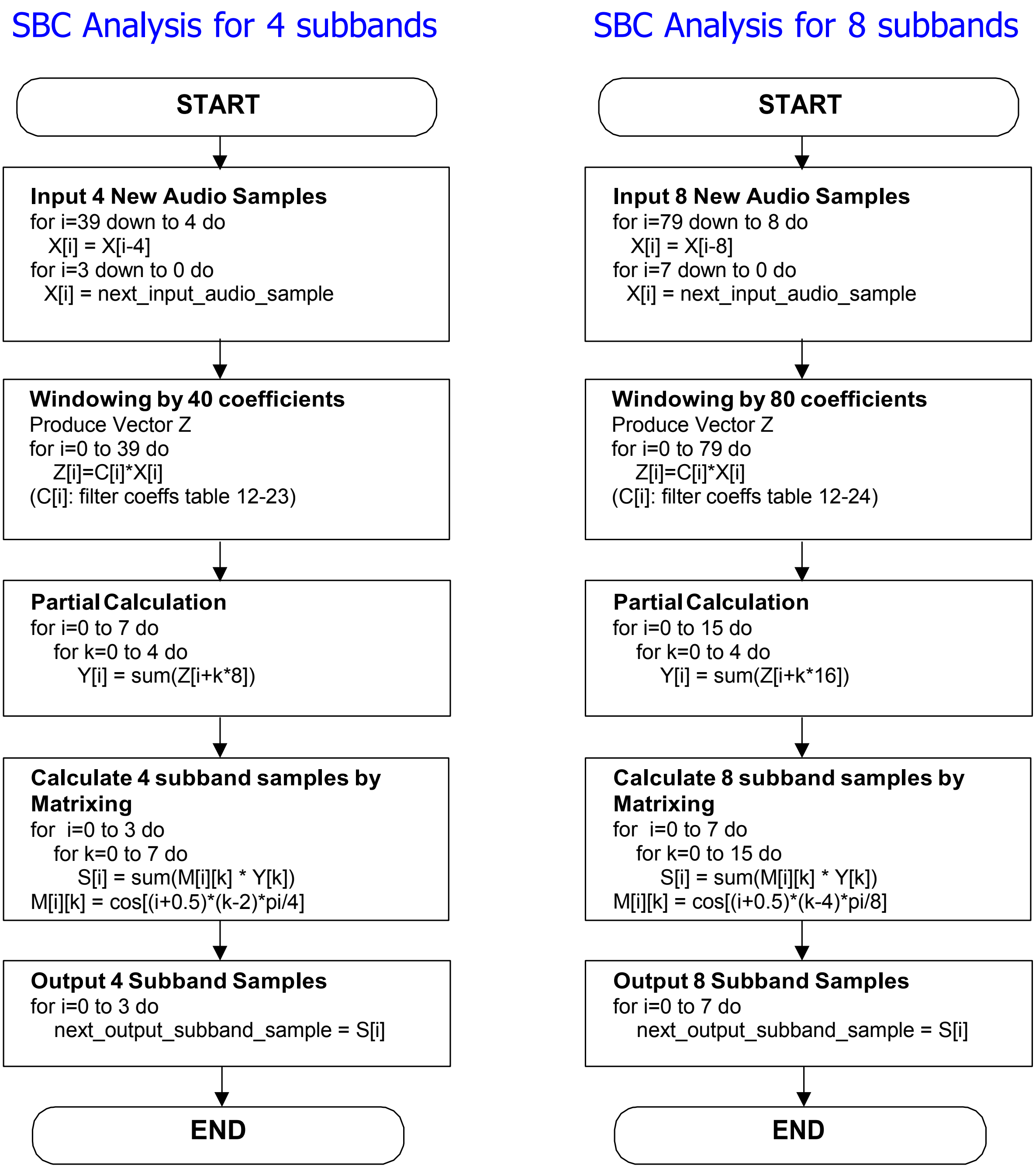
12.7.1 Analysis Filter

Analysis of the input PCM is calculated for each channel separately. For each block of nrof_subbands consecutive PCM samples the analysis filter is applied to calculate nrof_subbands subband samples. The analysis filter is a polyphase filterbank according to

$$h_m[n] = h_p[n]\cos\left(\left(m+\frac{1}{2}\right)\cdot\left(n-\frac{M}{2}\right)\cdot\frac{\pi}{M}\right),\quad m=[0,M-1], n=[0,L-1],$$

with M=nrof_subbands and L=10*nrof_subbands. The prototype filters for both M=4 and M=8 are in Section 12.8. This analysis filterbank has the same structure as the one that is used in [12].

A detailed filter block diagram can be found in Figure 12.5.



SBC Analysis for 4 subbands

```
Input 4 New Audio Samples
for i=39 down to 4 do
   X[i] = X[i-4]
for i=3 down to 0 do
  X[i] = next_input_audio_sample

Windowing by 40 coefficients
Produce Vector Z
for i=0 to 39 do
    Z[i]=C[i]*X[i]
(C[i]: filter coeffs table 12-23)

Partial Calculation
for i=0 to 7 do
    for k=0 to 4 do
        Y[i] = sum(Z[i+k*8])

Calculate 4 subband samples by Matrixing
for  i=0 to 3 do
    for k=0 to 7 do
        S[i] = sum(M[i][k] * Y[k])
M[i][k] = cos[(i+0.5)*(k-2)*pi/4]

Output 4 Subband Samples
for i=0 to 3 do
    next_output_subband_sample = S[i]
```

SBC Analysis for 8 subbands

```
Input 8 New Audio Samples
for i=79 down to 8 do
   X[i] = X[i-8]
for i=7 down to 0 do
  X[i] = next_input_audio_sample

Windowing by 80 coefficients
Produce Vector Z
for i=0 to 79 do
    Z[i]=C[i]*X[i]
(C[i]: filter coeffs table 12-24)

Partial Calculation
for i=0 to 15 do
    for k=0 to 4 do
        Y[i] = sum(Z[i+k*16])

Calculate 8 subband samples by Matrixing
for  i=0 to 7 do
    for k=0 to 15 do
        S[i] = sum(M[i][k] * Y[k])
M[i][k] = cos[(i+0.5)*(k-4)*pi/8]

Output 8 Subband Samples
for i=0 to 7 do
    next_output_subband_sample = S[i]
```

These Flow Diagrams are adapted from Figure C.4 and paragraph C.1.3 in ISO/IEC 11172-3

Figure 12.5: Flow Diagrams of the Analysis Filter

12.7.2 Scale Factors

For each subband a scale factor is calculated by taking the next higher scale factor value of the maximum absolute value in each subband. The scale factor values are defined in Section 12.6.2.

12.7.3 Joint_Stereo Channel Mode Operation

For the JOINT_STEREO channel mode operation a slightly different procedure is applied. From the L and R subband signals sum and difference subband signals are derived and scale factors are calculated for these sum and difference subband signals. A simple criterion may be used to determine whether the L and R subband signals are transmitted or the sum and difference subband signals, e.g., if the sum of the scale factors of L and R is larger than the sum of the scale factors of the sum and difference signals, the subband is coded using joint coding.

12.7.4 Bit Allocation

The bit allocation is exactly the same for the encoder and the decoder, and is described in Section 12.6.3.

12.7.5 Quantization

The subband samples are normalized and quantized using the following formula.

quantized_sb_sample[blk][ch][sb] = ⌊((sb_sample[blk][ch][sb] / scalefactor[ch][sb] + 1.0) * levels[ch][sb]) / 2.0 ⌋

12.8 Tables

In the case that the LOUDNESS bit allocation method is used in the bit allocation process, the next two tables are used.

offset4	fs = 16000	fs = 32000	fs = 44100	fs = 48000
sb = 0	-1	-2	-2	-2
sb = 1	0	0	0	0
sb = 2	0	0	0	0
sb = 3	0	1	1	1

Table 12.21: Offset table for four subbands

offset8	fs = 16000	fs = 32000	fs = 44100	fs = 48000
sb = 0	-2	-3	-4	-4
sb = 1	0	0	0	0
sb = 2	0	0	0	0
sb = 3	0	0	0	0
sb = 4	0	0	0	0
sb = 5	0	0	0	0
sb = 6	0	1	1	1
sb = 7	1	2	2	2

Table 12.22: Offset table for eight subbands

For the analysis and synthesis filters the filter coefficients are defined in the next two tables. The tables shall be read row-wise.

Proto_4_40			
0.00000000E+00	5.36548976E-04	1.49188357E-03	2.73370904E-03
3.83720193E-03	3.89205149E-03	1.86581691E-03	-3.06012286E-03
1.09137620E-02	2.04385087E-02	2.88757392E-02	3.21939290E-02
2.58767811E-02	6.13245186E-03	-2.88217274E-02	-7.76463494E-02
1.35593274E-01	1.94987841E-01	2.46636662E-01	2.81828203E-01
2.94315332E-01	2.81828203E-01	2.46636662E-01	1.94987841E-01
-1.35593274E-01	-7.76463494E-02	-2.88217274E-02	6.13245186E-03
2.58767811E-02	3.21939290E-02	2.88757392E-02	2.04385087E-02
-1.09137620E-02	-3.06012286E-03	1.86581691E-03	3.89205149E-03
3.83720193E-03	2.73370904E-03	1.49188357E-03	5.36548976E-04

Table 12.23:Filter coefficients for four subbands

Proto_8_80			
0.00000000E+00	1.56575398E-04	3.43256425E-04	5.54620202E-04
8.23919506E-04	1.13992507E-03	1.47640169E-03	1.78371725E-03
2.01182542E-03	2.10371989E-03	1.99454554E-03	1.61656283E-03
9.02154502E-04	-1.78805361E-04	-1.64973098E-03	-3.49717454E-03
5.65949473E-03	8.02941163E-03	1.04584443E-02	1.27472335E-02
1.46525263E-02	1.59045603E-02	1.62208471E-02	1.53184106E-02
1.29371806E-02	8.85757540E-03	2.92408442E-03	-4.91578024E-03
-1.46404076E-02	-2.61098752E-02	-3.90751381E-02	-5.31873032E-02
6.79989431E-02	8.29847578E-02	9.75753918E-02	1.11196689E-01
1.23264548E-01	1.33264415E-01	1.40753505E-01	1.45389847E-01
1.46955068E-01	1.45389847E-01	1.40753505E-01	1.33264415E-01
1.23264548E-01	1.11196689E-01	9.75753918E-02	8.29847578E-02
-6.79989431E-02	-5.31873032E-02	-3.90751381E-02	-2.61098752E-02
-1.46404076E-02	-4.91578024E-03	2.92408442E-03	8.85757540E-03
1.29371806E-02	1.53184106E-02	1.62208471E-02	1.59045603E-02
1.46525263E-02	1.27472335E-02	1.04584443E-02	8.02941163E-03
-5.65949473E-03	-3.49717454E-03	-1.64973098E-03	-1.78805361E-04
9.02154502E-04	1.61656283E-03	1.99454554E-03	2.10371989E-03
2.01182542E-03	1.78371725E-03	1.47640169E-03	1.13992507E-03
8.23919506E-04	5.54620202E-04	3.43256425E-04	1.56575398E-04

Table 12.24: Filter coefficients for eight subbands

12.9 Calculation of Bit Rate and Frame Length

Bit Rate (**bit_rate**) is calculated using the following equation:

$$\mathbf{bit\_rate} = 8 * \mathbf{frame\_length} * \mathbf{f_s} / \mathbf{nrof\_subbands} / \mathbf{nrof\_blocks},$$

where $\mathbf{f_s}$, **nrof_subbands** and **nrof_blocks** denote sampling frequency, number of subbands and number of blocks, respectively. Bit Rate is expressed in kb/s, because $\mathbf{f_s}$ is expressed in kHz. The Frame Length (**frame_length**) is expressed in bytes as

$$\begin{aligned}\mathbf{frame\_length} = 4 + (4 * \ \mathbf{nrof\_subbands} * \mathbf{nrof\_channels}\ ) / 8 \\ + \lceil \mathbf{nrof\_blocks} * \mathbf{nrof\_channels} * \mathbf{bitpool} / 8 \rceil.\end{aligned}$$

for the MONO and DUAL_CHANNEL channel modes, and

$$\textbf{frame\_length} = 4 + (4 * \textbf{nrof\_subbands} * \textbf{nrof\_channels}) / 8 \\ + \lceil (\textbf{join} * \textbf{nrof\_subbands} + \textbf{nrof\_blocks} * \textbf{bitpool}) / 8 \rceil.$$

for the STEREO and JOINT_ STEREO channel modes.

Here, **nrof_channels** and **bitpool** denote number of channels and bitpool value, respectively. When joint stereo is used, **join** = 1, otherwise 0. For reference, see Section 12.5.

13 Appendix C (Informative): Signalling Flows

This section contains an example of typical signalling procedures defined in AVDTP for audio streaming set up. This section is informative only. For details, refer to GAVDP[3] and AVDTP[4]. In this example, the **SRC** is assumed to be the **INT**, while the **SNK** to be the **ACP**.

13.1 Audio Streaming Set Up

The initial states of the both devices are <IDLE>.

The **SRC** initiates *Stream Endpoint (SEP) Discovery* procedure. This procedure serves to return the media type and SEID for each stream end-point. The **SRC** finds the audio-type stream end-point.

Then, *Get Capabilities* procedure is initiated to collect service capabilities of the **SNK**. There are two kinds of service capabilities; one is an application service capability and the other is a transport service capability. The application service capability for A2DP consists of audio codec capability and content protection capability. Regarding the transport service capability, refer to Section 5.4 in AVDTP[4].

Based on collected SEP information and service capabilities, the **SRC** determines the most suitable audio streaming parameters (codec, content protection and transport service) for the **SNK** and the **SRC** itself. Then, **SRC** requests the **SNK** to configure the audio parameters of the **SNK** by using the *Stream Configuration* procedure. The **SRC** also configures the audio parameters of itself.

Then, L2CAP channels are established as defined in the *Stream Establishment* procedure. Finally, the states of both devices are set at <OPEN>.

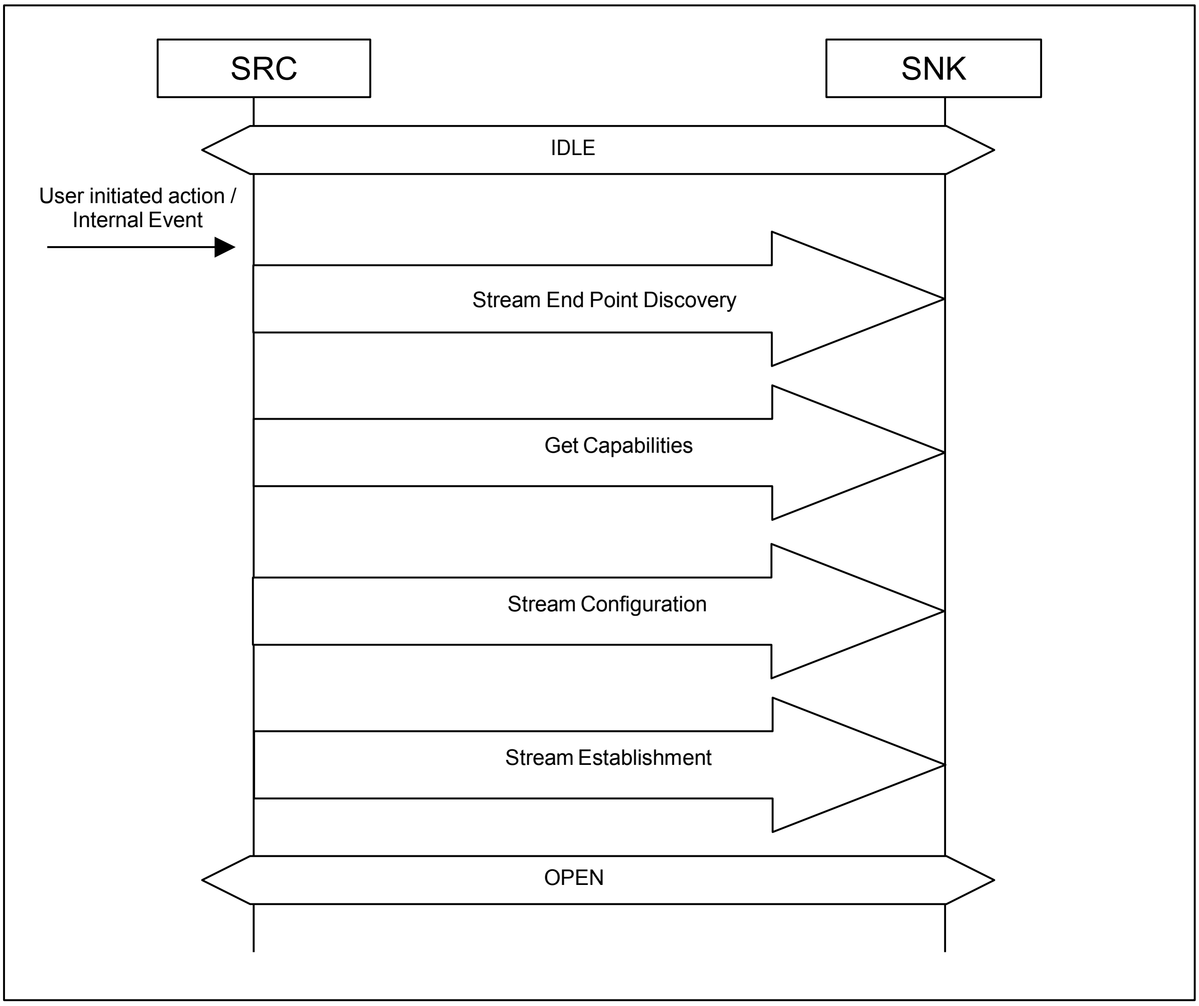


Figure 13.1: Audio Streaming Set Up

13.2 Audio Streaming

The **SRC** initiates *Start Streaming* procedure by a user initiated action or an internal event. The states of both devices are changed from <OPEN> to <STREAMING>. Audio streaming is started after this procedure is completed.

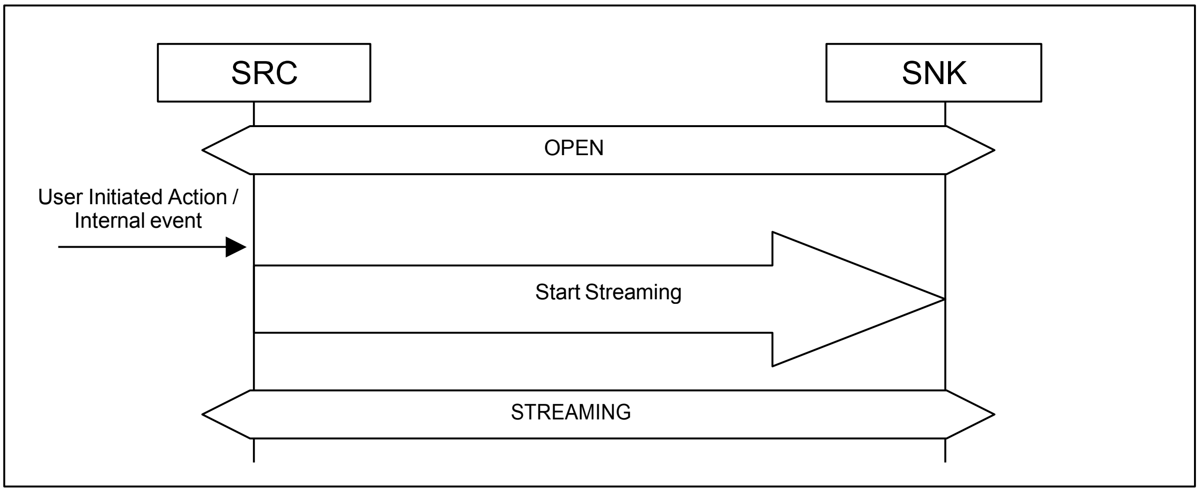


Figure 13.2: Audio Streaming

14 Appendix D: Acronyms and Abbreviations

Acronym	Description
A/V	Audio/Video
A2DP	Advanced Audio Distribution Profile
ACP	Acceptor
AVDTP	Audio/Video Distribution Transport Protocol
AVRCP	Audio/Video Remote Control Profile
CP_Type	Content Protection Type
CRC	Cyclic Redundancy Check
GAP	Generic Access Profile
GAVDP	Generic Audio/Video Distribution Profile
ICS	Implementation Conformance Statement
IETF	Internet Engineering Task Force
INT	Initiator
LM	Link Manager
LSB	Least Significant Bit (Byte)
MPEG	Moving Picture Expert Group
MSB	Most Significant Bit (Byte)
MTU	Maximum Transmission Unit
PSM	Protocol/Service Multiplexer
QoS	Quality of Service
RFA	Reserved for Future Additions
RFD	Reserved for Future Definition
RTP	Real-time Transport Protocol
SBC	Low Complexity Subband Codec
SDP	Service Discovery Protocol
SEID	Stream End Point Identifier
SEP	Stream End Point
SNK	Sink
SRC	Source
TSS	Test Suite Structure
VDP	Video Distribution Profile