SMPTE STANDARD

SMPTE 362M-2002

for Television —
Object Data Format for the Exchange of DV-Based Audio, Data and Compressed Video using ATM Common Layer over Asynchronous Transfer Mode (ATM) AAL Type 1

1 Scope

This standard defines the object data format for the exchange of DV-based audio, data, and compressed video (which data structure is defined in SMPTE 314M) over the common layer defined in SMPTE 354M. The standard covers the transmission of audio, compressed video, and subcode data with DV-based 25-Mb/s and 50-Mb/s data structures for 525/60 and 625/50 systems and the transmission of the header data of serial data transport interface defined in SMPTE 305.2M.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

SMPTE 305.2M-2000, Television — Serial Data Transport Interface (SDTI)

SMPTE 314M-1999, Television — Data Structure for DV-Based Audio, Data and Compressed Video — 25 and 50 Mb/s

SMPTE 354M, Television — ATM Common Layer for Transport of Packetized Audio, Video and Data over Asynchronous Transfer Mode using ATM Adaptation Layer Type 1

3 Common layer

As defined in SMPTE 354M, the common layer is the next higher layer of the AAL1 and provides transmission packets of applications. The common layer consists of two layers: the lower layer is the SYNC layer and the higher layer is the container layer.

3.1 SYNC layer

The SYNC stream block (SSB), which is defined in the SYNC layer, includes containers. The SSB has one or more containers, and the transmission order of the SSB is defined in SMPTE 354M.

3.2 Container layer

The container layer defines the structure of the container in the SSB. Each container is intended to hold a video frame (optionally two fields) worth of packetized audio, video, and data.

3.2.1 Container format

The container has two basic modes: the simple mode and the extended header mode. The simple mode supports four objects of several restricted types. The extended header mode supports up to 16 objects.

3.2.2 Object classification system

As described in SMPTE 354M, the container system provides a mechanism for defining the bit and byte packing of audio-video data types. Object classification is represented in the header by the object class word. This word is composed of two key elements: the type byte and the index byte object. The type byte is a coarse classification of the data type, and the index type is the key into the table (which is defined in figure 6 of SMPTE 354M) which represents data types.

Approved
January 31, 2002

3.2.3 Compressed AV stream

Video program is defined as an object class 20_h and *compressed AV stream (dv-based)* is defined as table index 0001_h of the video program object in SMPTE 354M. The format of a compressed AV stream is based on the subcontainer format shown in figure 1.

The subcontainer is composed of a 12-byte stream header and multiple/single compressed data stream (CDS) packets.

3.2.3.1 Stream header

The stream header, which is shown in figure 1, is composed as follows:

Stream descriptor: 32 bits (4 bytes)
Stream time stamp: 32 bits (4 bytes)
CDS packet length: 32 bits (4 bytes)

3.2.3.2 Stream descriptor

The stream descriptor (which is shown in figure 2) is composed of the following:

Packet type: 16 bits (2 bytes)
Stream type: 16 bits (2 bytes)

3.2.3.2.1 Packet type

The packet type is 16-bit data and indicates the CDS packet structure. The packet type is defined as follows:

– Bit 31: The existence indication bit of the packet time stamp word.

0 - indicates that the packet time stamp word exists.
1 - indicates that the packet time stamp word does not exist.

– Bit 30: The existence indication bit of the byte count word.

0 - indicates that the byte count word exists.
1 - indicates that the byte count word does not exist.

– Bit 29: The existence indication bit of the FEC (forward error correction) word.

0 - indicates that the FEC word exists.
1 - indicates that the FEC word does not exist.

– Bit 28: The indication bit that indicates multiple packets or single packet.

0 - indicates that the subcontainer is composed of a single CDS packet.
1 - indicates that the subcontainer is composed of multiple CDS packets.

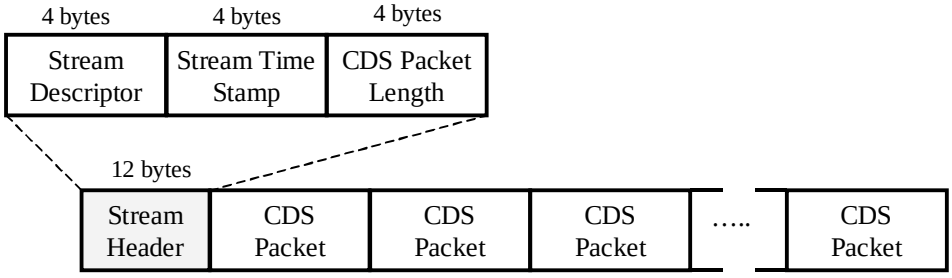


Figure 1 – Compressed AV stream format

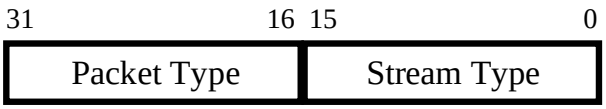


Figure 2 – Stream descriptor

– Bit 27: The indication bit that indicates that the stream time stamp word is valid or invalid.

0 - indicates that the stream time stamp word is valid.
1 - indicates that the stream time stamp word is invalid.

– Bit 26: The indication bit that indicates fixed-packet size or variable-packet size.

0 - indicates that the CDS packet size is fixed.
1 - indicates that the CDS packet size is variable and the CDS packet length word is invalid.

– Bit 25 and Bit 24: Reserved for future purposes (default value is hex 00).

– Bit 23 through Bit 16: Indicates the byte length of FEC.

3.2.3.2.2 Stream type

The stream type is 16-bit data. The definition of the stream type depends on the compressed data stream.

3.2.3.3 Stream time stamp

The stream time stamp is composed of a 32-bit packet length word and indicates the point of time at which the compressed AV stream object is multiplexed into the container. The stream time stamp is used for recovering the relative stream object time from end to end and is valid when B27 of the packet type is set to 0. The synchronization scheme using this word is described later.

3.2.3.4 CDS packet length

The CDS packet length is composed of a 32-bit packet length word and indicates the byte length of the CDS packet multiplexed immediately after the stream header. The CDS packet length word is valid when B27 of the packet type is set to 0.

3.2.3.5 CDS packet format

The CDS packet format is composed of a 32-bit packet time stamp word, a 32-bit byte count word, an N byte CDS packet payload, and M bytes FEC. The CDS packet format is shown in figure 3.

3.2.3.5.1 CDS packet time stamp

The CDS packet time stamp is a 32-bit word and exists when B31 of the packet type is set to 0. The CDS packet time stamp is used for recovering the relative CDS packet timing from end to end. The reference clock for the packet time stamp is the same frequency as that for the object time stamp in the container header. The synchronization scheme using this word is described later.

3.2.3.5.2 Byte count

The byte count is a 32-bit word and exists when B30 of the packet type is set to 0. The byte count indicates the byte length of the variable CDS packet payload size or the valid byte length in the fixed CDS packet payload. If the packet time stamp word does not exist, the byte count word is located at the head word of the CDS packet.

3.2.3.5.3 CDS packet payload

The CDS packet payload format depends on the compressed data stream.

3.2.3.5.4 FEC (forward error correction)

The forward error correction is M byte code and exists when B29 of the packet type is set to 0. The FEC applies to the first byte through the last byte of the CDS packet.

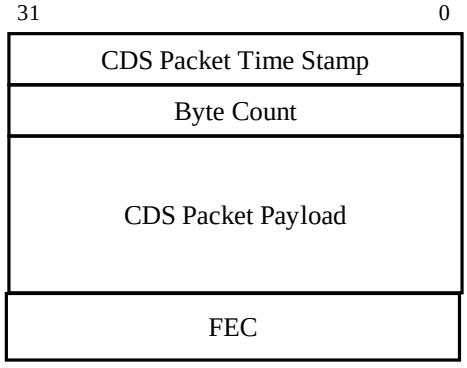


Figure 3 – CDS packet format

3.2.3.6 Synchronization scheme

In a frame/field compression system, such as DV-based audio, data and compressed video, if only the frame/field frequency is recovered accurately, it is not necessary to recover the relative CDS packet timing accurately but to recover the CDS packet within the frame/field in which they were generated by the sender. Therefore, the stream time stamp (STS) or the CDS packet time stamp (CTS) is not always necessary.

4 Object for SDTI information and DV-based audio, data and compressed video

4.1 Defined objects

Two kinds of objects, the object for SDTI information and the object for DV-based audio, data, and compressed video are defined and shown in figure 4.

4.2 Object format for SDTI information

The object for SDTI information (SDTI header is defined in SMPTE 305.2M) is treated as the ancillary data type object and is shown in figure 5. The line number information in the object for SDTI information shall be the same as the start line number of each video frame in the SDTI channel unit. The object for SDTI information shall correspond to data which are B0, B1, B2, B3, B4, B5, B6, and B7 (8 bits) of each word numbered from 3 to 52 in the SDTI header (which is defined in SMPTE 305.2M). The data length is 10 bits; B0 through B9. B9 is the most significant bit (MSB).

4.3 Object format for DV-based audio, data and compressed video

DV-based audio, data, and compressed video data are treated as the CDS packet payload of the video program object.

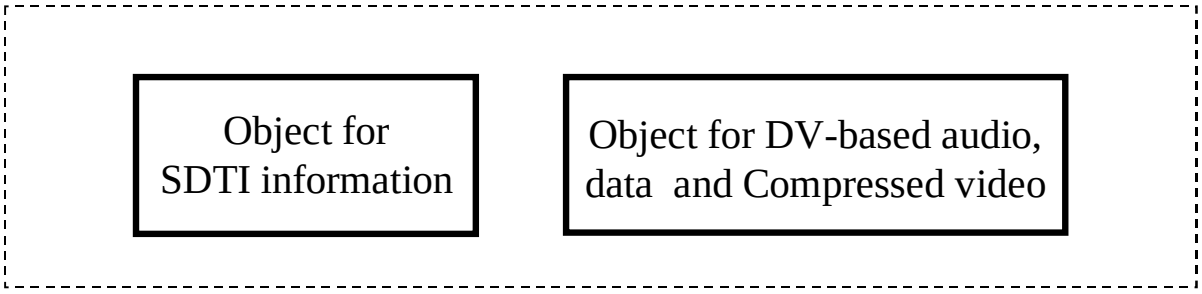


Figure 4 – Two kinds of defined objects

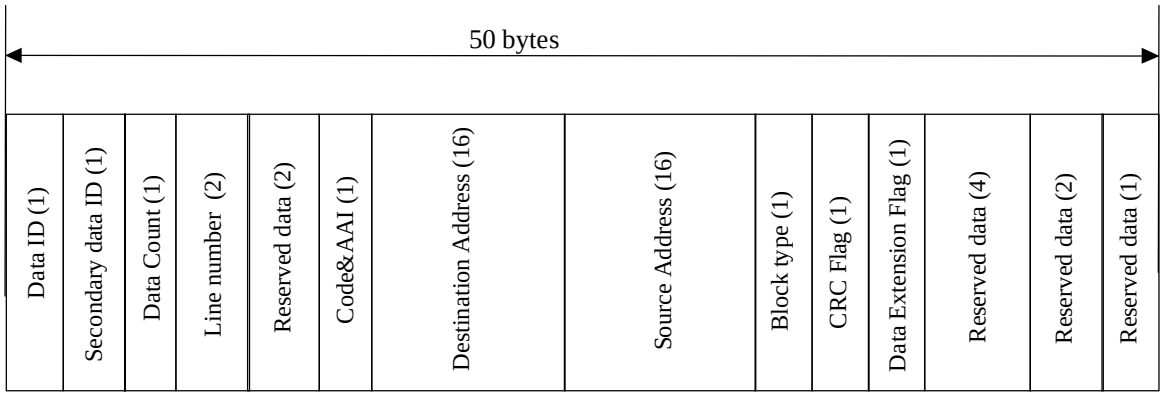


Figure 5 – Object for SDTI information

4.3.1 Stream header

4.3.1.1 Stream descriptor

4.3.1.1.1 Packet type

– Bit 31 = 1: The packet time stamp word does not exist.

– Bit 30 = 1: The byte count word does not exist.

– Bit 29 = 0 or 1: According to the error scheme the DV-based stream transport adopts.

– Bit 28 = 1: The subcontainer is composed of multiple CDS packets.

– Bit 27 = 0 or 1: According to the relationship to other objects in the container.

– Bit 26 = 0: The CDS packet size is fixed.

– Bit 25 and bit 24: Default value binary 00.

– Bit 23 through bit 16: When the 4-byte FEC word exists, the length of FEC shall have the binary value 04. When the FEC word does not exist, the length of FEC shall have the binary value 00.

4.3.1.1.2 Stream type

– Bit 15 through bit 8: Reserved for future purposes (default value is $0000\ 0000_b$).

– Bit 7: Reserved for future purposes (default value is 0_b).

– Bit 6 through bit 4: These three bits indicate the DIF structure with the following values:

Bit 6	Bit 5	Bit 4	
0	0	0	: Reserved
0	0	1	: Reserved
0	1	0	: Reserved
0	1	1	: 25-Mb/s structure
1	0	0	: Reserved
1	0	1	: 50-Mb/s structure
1	1	0	: Reserved
1	1	1	: Reserved

– Bit 3 through bit 0: Reserved for future purposes (default value is 0000b).

4.3.1.1.3 CDS packet length

When the FEC word exists, the packet length shall have the value hex 00 00 00 A4 (164 bytes).

When the FEC word does not exist, the packet length shall have the value hex 00 00 00 A0 (160 bytes).

4.3.1.1.4 CDS packet format

According to the packet type, the CDS packet format of the DV-based compression stream is shown in figure 6.

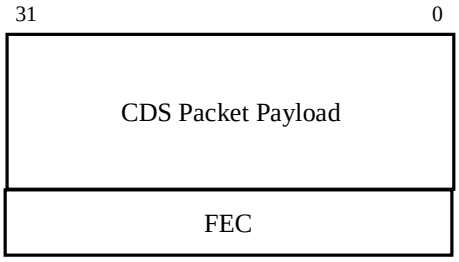


Figure 6 – CDS packet format

4.3.1.1.5 CDS packet payload

The CDS packet payload format for the DV-based compression stream is shown in figure 7. The DIF block ID (ID0-2) and DIF block data are specified in SMPTE 314M. A CDS packet includes two DIF block data. In the 525/60 system, one compressed AV stream object is composed of 750 CDS packets (1500 DIF blocks) for 25-Mb/s structure or 1500 CDS packets (3000 DIF blocks) for 50-Mb/s structure. In the 625/50 system, one compressed AV steam object is composed of 900 (1800 DIF blocks) for 25-Mb/s or 1800 CDS packets (3600 DIF blocks) for 50-Mb/s structure.

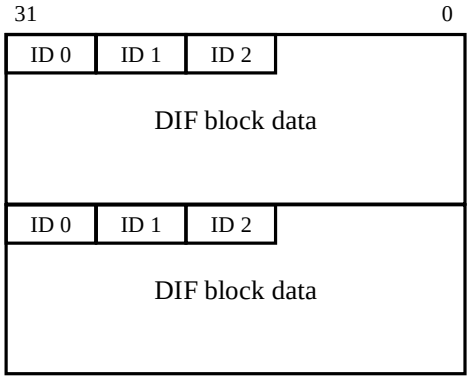


Figure 7 – CDS packet payload format

4.3.1.1.6 FEC (forward error correction)

The CDS packet data are optionally protected by the forward error correction (FEC). The FEC is a 4-byte code and is located immediately after the CDS packet payload when B29 of the packet type is set to 0. The FEC applies to the first byte through the last byte of the CDS packet. A 4-byte Reed Solomon error correction shall be used as the method of FEC.

FEC is not necessary when only the ATM AAL type 1 is used as the transmission means between end to end. However, for example, when the FC-AV transmission means with FEC option is used together with the ATM AAL type 1 transmission means between end to end, FEC is used to make the gateway between the FC-AV network and the ATM network simpler.

4.3.2 Object data format

When FEC does not exist, the object as the compressed AV stream for DV-based audio, data, and compressed video data are shown in figures 8 and 9.

The object for DV-based audio, data, and compressed video data for 25 Mb/s is as follows: (figure 8):

DIF frame data (DIF blocks):

1,500 DIF blocks (120,000 bytes) (525/60)
1,800 DIF blocks (144,000 bytes) (625/50)

The object for DV-based audio, data, and compressed video data for 50 Mb/s is as follows (figure 9):

DIF frame data (DIF blocks):

3,000 DIF blocks (240,000 bytes) (525/60)
3,600 DIF blocks (288,000 bytes) (625/50)

4.3.3 DIF frame data

The DIF frame data shall consist of 1,500 DIF blocks (525/60) or 1,800 DIF blocks (625/50), and each DIF block consists of 80 bytes. The DIF block is specified in SMPTE 314M. The transmission order within one compressed video frame for 25 Mb/s and 50 Mb/s structures is shown in figures 10 and 11.

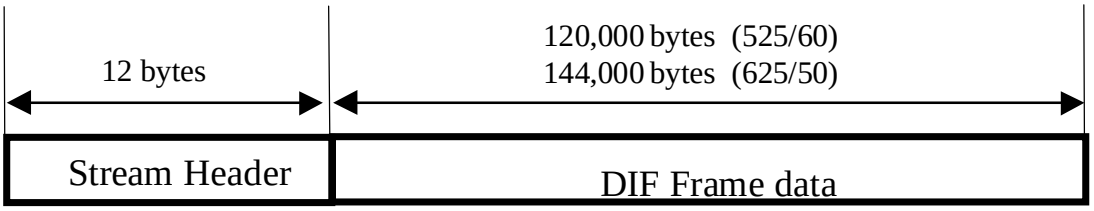


Figure 8 – Example of object for DV-based audio, data and compressed video for 25 Mb/s

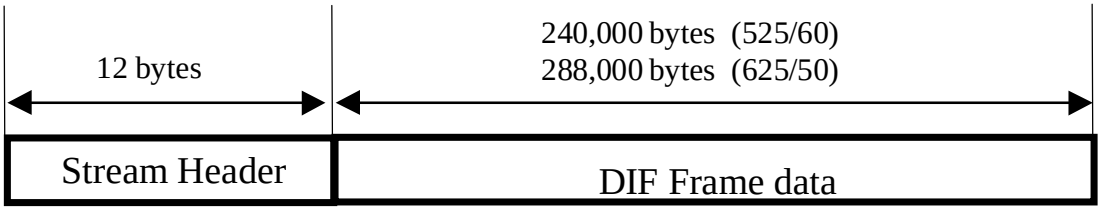
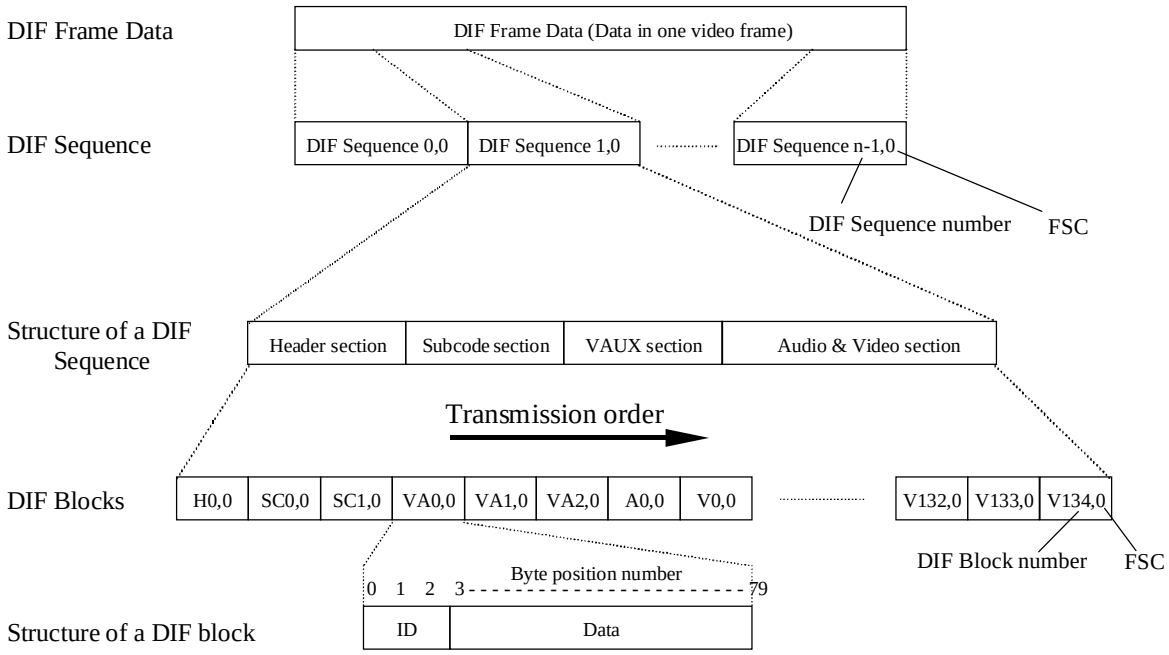
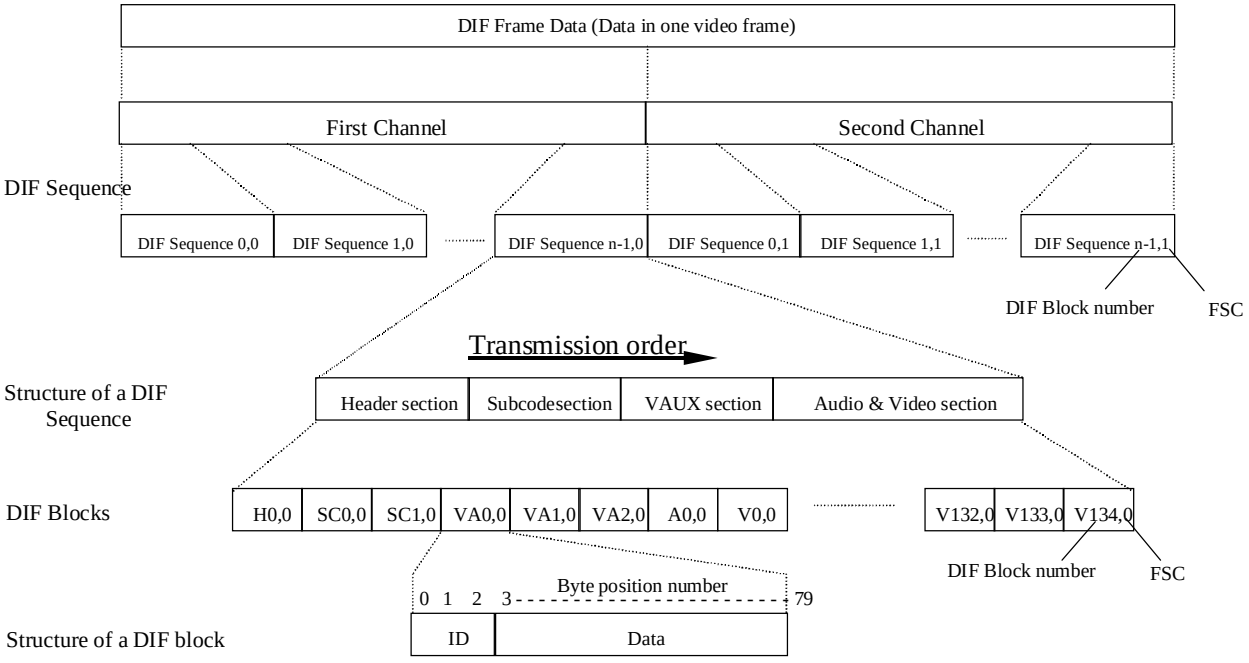


Figure 9 – Example of object for DV-based audio, data and compressed video for 50 Mb/s



NOTE – n = 10 for 525/60 system and n = 12 for 625/50 system. FSC = first/second channel.

Figure 10 – DIF frame data structure for 25 Mb/s



NOTE – n = 10 for 525/60 system and n = 12 for 625/50 system. FSC = first/second channel.

Figure 11 – DIF frame data structure for 50 Mb/s

4.4 Container format for DV-based audio, data and compressed video with SDTI information

The container format for DV-based audio, data, and compressed video is shown in figure 12. In figure 12, there are two objects: one is the SDTI information, and the other is DV-based one-frame data. The object type and the index for the object type are described in SMPTE 354M (figure 6 and table 2).

The SDTI information is defined as an ancillary data object (object class type 50_h) and the object 0 of the containers is assigned. DV-based audio, data, and compressed video is defined as a compressed AV stream of the video program object (object class type 20_h, table index 0001_h) and the object 2 of the containers is assigned. For DV-based audio, data, and compressed video transmission, two DIF blocks are assigned to each CDS packet.

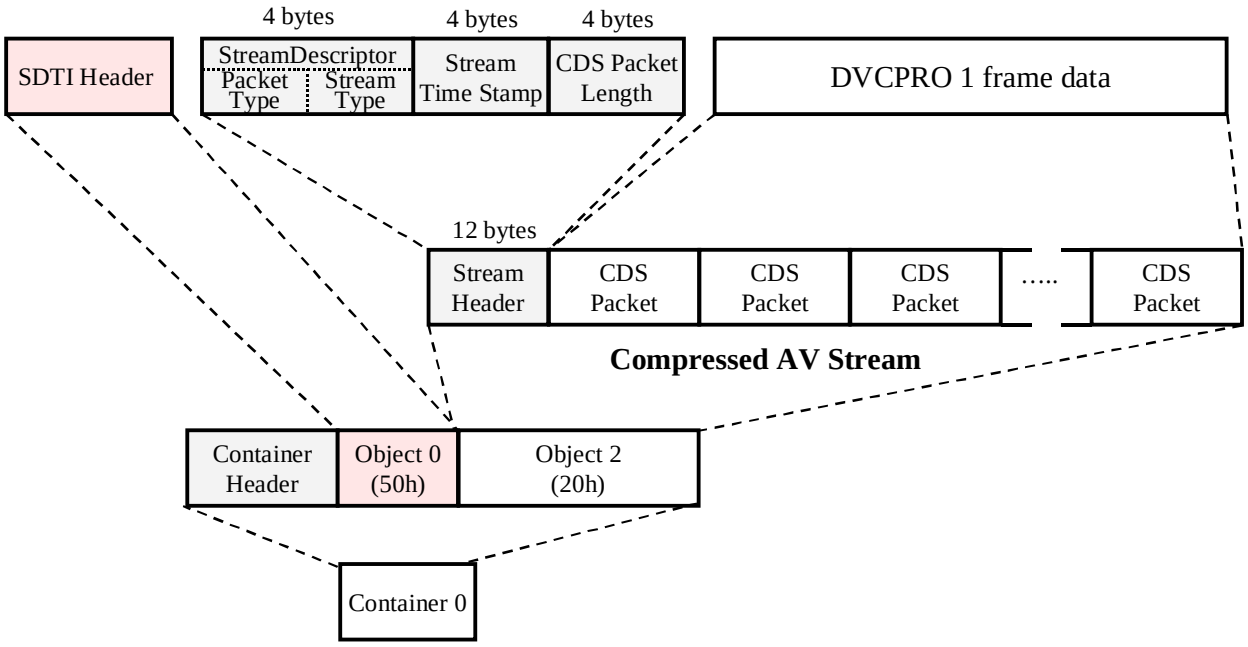


Figure 12 – Example of container format for normal speed transmission

Annex A (informative)
SSB format examples for DV-based audio, data and compressed video

A.1 SSB format for normal speed transmission of a single program

An example of the SSB format for normal speed transmission of one DV-based audio, data, and compressed video program is shown in figure A.1. In figure A.1, there is one container in one SSB, and the container contains one video frame worth of video, audio, and ancillary data. It is SDTI information and DV-based audio, data, and compressed video of one frame that correspond to the data of one video frame worth of video, audio, and ancillary data. In each container, SDTI information is used as object 0, and one frame of DV-based audio, data, and compressed video is used as object 2.

In figure A.1, the value of the SSB value length is as follows:

{The value of the SSB value length} = 4 + 1 + 1 + 9 + {The value of container 0 size} bytes.

For 25 Mb/s DV-based audio, data, and compressed video, the value of the object size in one frame is 120,012 bytes for 525/60 and 144,012 bytes for 625/50. For 50 Mb/s DV-based audio, data, and compressed video data, the value of the object size in one frame is 240,012 bytes for 525/60 and 288,012 bytes for 625/50.

For example, for 25-Mb/s DV-based when the FEC word does not exist in the CDS packet (when bit 29 of the stream

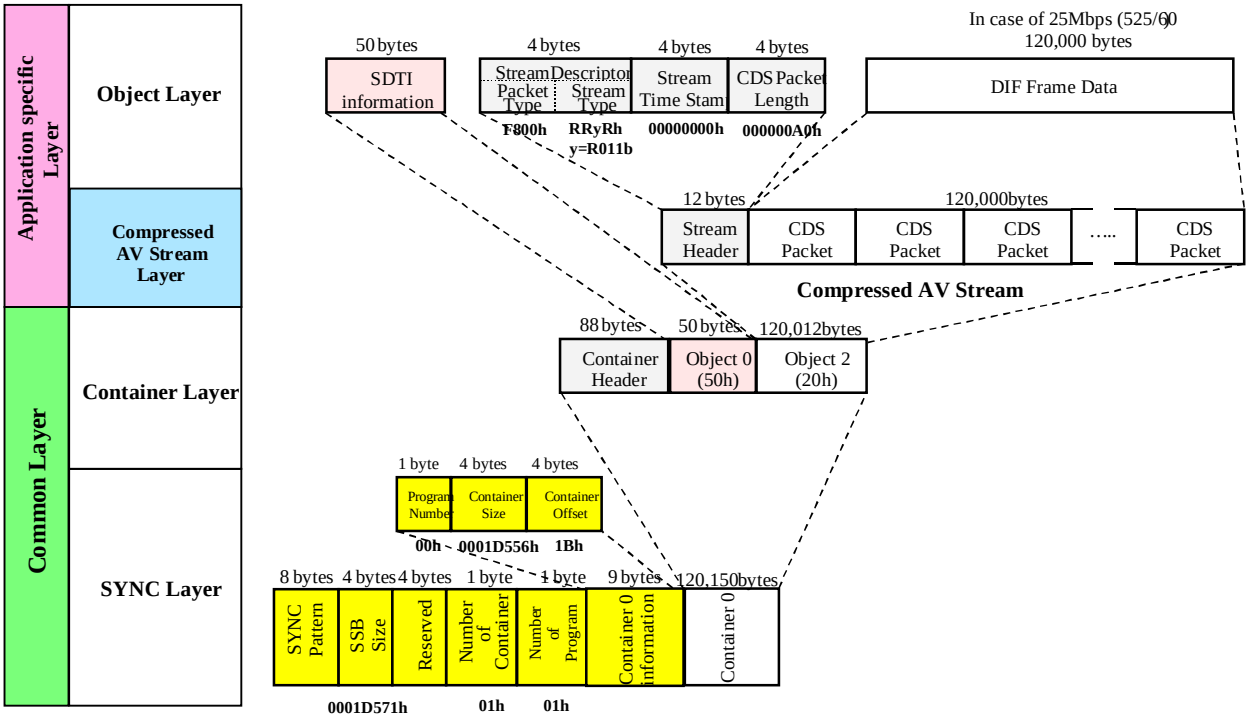


Figure A.1 – Example of the SSB format for normal speed transmission of one DV-based audio, data and compressed video

header = 1), the total length of the CDS packet in one object (object 2) becomes equal to the length of DV-based 1-frame data. Therefore, the value of container 0 size for 25 Mb/s DV-based 525/60 in one frame is as follows:

{The value of container 0 size for 25 Mb/s DV-based 525/60} = 88 + 50 + 120,012 = 120,150 bytes.

The value of the SSB value length for 25 Mb/s DV-based 525/60 in one frame is as follows:

{The value of the SSB value length for 25 Mb/s DV-based 525/60 in one frame} = 15 +120,150 = 120,165 bytes.

The value of container 0 offset is as follows:

{The container 0 offset} = 35 bytes.

Regarding the SSB header, the number of the container, the number of the program, and the container information which includes the program number, the value of the container 0 size, and the value of the container 0 offset are as follows:

SSB value length: $1D565_h$ ($129,165_d$)
Number of container: 01_h
Number of program: 01_h

Container 0 information
Program number: 01_h
Container 0 size: 32_h (50_d)
Container 0 offset: 58_h (88_d)

Regarding the compressed AV stream, the value of the packet type in the stream descriptor becomes 1111 1000 0000 0000_b; that is, $F800_h$, in the following cases:

When the packet time stamp word does not exist, bit 31 is set to 1.

When the byte count word does not exist, bit 30 is set to 1.

When the FEC word does not exist, bit 29 is set to 1.

When the subcontainer is composed of multiple CDS packets, bit 28 is set to 1.

When the stream time stamp word is invalid, bit 27 is set to 1.

When the CDS packet size is fixed, bit 26 is set to 0.

Bit 25 and bit 24 are 00_b, and bit 23 through bit 16 are all set to 0.

The value of the stream type in the stream descriptor becomes 0000 0000 0011 0000_b (0030_h) in the following cases:

When bit 15 through bit 8 are set to default value, 0000 0000_b.

When bit 7 is set to default value, 0_b.

When 25-Mb/s DV-based 525/60, bit 6 through bit 4 are set to 011_b.

When bit 3 through bit 0 are set to default value 0000_b.

Therefore, the value of the stream descriptor is F800 0030_h.

The stream time stamp is set to 0000 0000_h because it does not exist.

The CDS packet length, when ECC does not exist, becomes the same as the length of two DIF blocks, 160 bytes (0000 $00A0_h$ bytes).

The value of the container header is shown in figure A.2. The value of the sequence number is defined by the user. The value of the clip ID is set to 0000 0000_h as the default value (null). The value of the container time stamp is set to 0000 0000 0000 0000_h as the default value (null). The value of the video frame rate is set to 05_h as is 29.97 NTSC. The value of the transmission rate is set to 01_h as the normal transmission rate. The mode is simple mode (bit 7= 0_b), so the value of the mode is set to 00_h. The object number is set to 02_h for object 0 and object 2. The value of the object type defined is set to 00_h as the default value (null).

A.2 SSB format for normal speed transmission of four programs

An example of the SSB format for normal speed transmission of four DV-based audio, data, and compressed video programs is shown in figure A.3. In figure A.3, there are four containers in one SSB, and each container contains one video frame worth of video, audio, and ancillary data. The

Word	Identifier	BYTE 0	BYTE 1	BYTE 2	BYTE 3
0	Sequence number	xx_h	xx_h	xxh	xxh
1	Clip ID	00_h	00_h	00h	00h
2	Container time stamp	00_h	00h	00h	00h
3	Container time stamp	00_h	00h	00h	00h
4	Transmission type	05_h	01h	00h	00h
5	Container type	00_h	02h	00h	00h
6	Object 0 class	50_h	02h	00h	00h
7	Object 0 size	00_h	00h	00h	32h
8	Object 0 offset	00_h	00h	00h	58h
9	Object 0 Object type defined	00_h	00h	00h	00h
10	Object 1 class	00_h	00h	00h	00h
11	Object 1 size	00_h	00h	00h	00h
12	Object 1 offset	00_h	00h	00h	00h
13	Object 1 Object type defined	00_h	00h	00h	00h
14	Object 2 class	20_h	00h	00h	01h
15	Object 2 size	00_h	01h	D4h	C0h
16	Object 2 offset	00_h	00h	00h	8Ah
17	Object 2 Object type defined	00_h	00h	00h	00h
18	Object 3 class	00_h	00h	00h	00h
19	Object 3 size	00_h	00h	00h	00h
20	Object 3 offset	00_h	00h	00h	00h
21	Object 3 Object type defined	00_h	00h	00h	00h

Figure A.2 – Example of container header format for normal speed transmission of one DV-based audio, data and compressed video

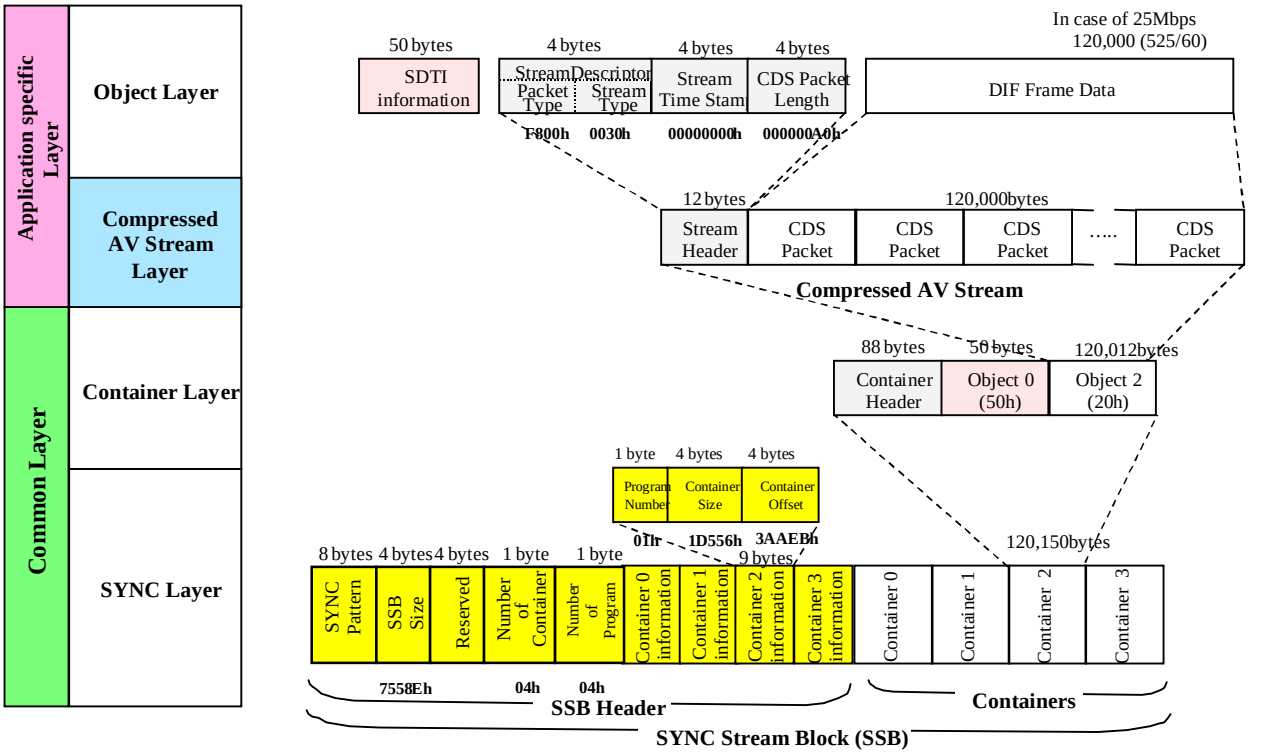


Figure A.3 – Example of the SSB format for normal speed transmission of four DV-based audio, data and compressed video

SDTI information and the DV-based audio, data, and compressed video of one frame that correspond to the data of one video frame worth of video, audio, and ancillary data. In each container, SDTI information is used as object 0, and one frame of DV-based audio, data, and compressed video is used as object 2.

In figure A.3, the value of the SSB value length is as follows:

{The value of the SSB value length} = 4 + 1 + 1 + (9 x 4)
+ {The value of container 0 size} + {The value of container 1 size}
+ {The value of container 2 size) + {The value of container 3 size} bytes

For 25 Mb/s DV-based audio, data, and compressed video, the value of object size in one frame is 120,012 bytes for 525/60 and 144,012 bytes for 625/50. For 50 Mb/s DV-based audio, data, and compressed video data, the value of object size in one frame is 240,012 bytes for 525/60 and 288,012 bytes for 625/50.

For example, for 25 Mb/s DV-based when the FEC word does not exist in the CDS packet (where bit 29 of the stream header = 1), the total length of the CDS packet in one object (object 2) becomes equal to the length of DV-based 1-frame data. Therefore, the value of container 0 size of 25 Mb/s DV-based 525/60 in one frame is as follows:

{The value of container 0 size for 25 Mb/s DV-based 525/60}
= {The value of container 1 size for 25 Mb/s DV-based 525/60}
= {The value of container 2 size for 25 Mb/s DV-based 525/60}
= {The value of container 3 size for 25 Mb/s DV-based 525/60}
= 88 + 50 + 120,012
= 120,150 bytes

The value of the SSB value length of 25 Mb/s DV-based 525/60 in one frame is as follows:

{The value of the SSB value length for 25 Mb/s DV-based 525/60 in one frame}
= 42 + 120,150*4
= 480,642 bytes

Container offset values are as follows:

{The value of container 0 offset} = 62 bytes
{The value of container 1 offset} = 62 + 120,150 = 120,212 bytes
{The value of container 2 offset} = 62 + (120,150*2) = 240,362 bytes
{The value of container 3 offset} = 62 + (120,150*3) = 360,512 bytes

Other parameter values in figure A.3 are set the same as those in annex A.1.

A.3 SSB format for four-times-faster-than-real-time transmission of one program

An example of the SSB format for four-times-faster-than-real-time transmission of one DV-based audio, data, and compressed video program is shown in figure A.4. In figure A.4, there are four containers in one SSB, and each container contains one video frame worth of video, audio, and ancillary data. The SDTI information and the DV-based audio, data, and compressed video of one frame correspond to the data of one video frame worth of video, audio, and ancillary data. In each container, SDTI information is used as object 0, and one frame of DV-based audio, data, and compressed video is used as object 2.

In figure A.4, the value of the SSB value length is as follows:

{The value of the SSB value length} = 4 + 1 + 1 + (9 x 4)
+ {The value of container 0 size} + {The value of container 1 size}
+ {The value of container 2 size} + {The value of container 3 size} bytes

For 25 Mb/s DV-based audio, data, and compressed video, the value of object size in one frame is 120,012 bytes for 525/60 and 144,012 byes for 625/50. For 50 Mb/s DV-based audio, data, and compressed video data, the value of object size in one frame is 240,012 bytes for 525/60 and 288,012 bytes for 625/50.

For example, for 25 Mb/s DV-based when the FEC word does not exist in the CDS packet (where bit 29 of the stream header = 1), the total length of the CDS packet in one object (object 2) becomes equal to the length of DV-based 1-frame data. Therefore, the value of container 0 size of 25 Mb/s DV-based 525/60 in one frame is as follows:

{The value of container 0 size for 25 Mb/s DV-based 525/60}
= {The value of container 1 size for 25 Mb/s DV-based 525/60}
= {The value of container 2 size for 25 Mb/s DV-based 525/60}
= {The value of container 3 size for 25 Mb/s DV-based 525/60}
= 88 + 50 + 120,012
= 120,150 bytes

The value of the SSB value length of 25 Mb/s DV-based 525/60 in one frame is as follows:

{The value of the SSB value length for 25 Mb/s DV-based 525/60 in one frame}
= 42 + 120,150*4
= 480,642 bytes

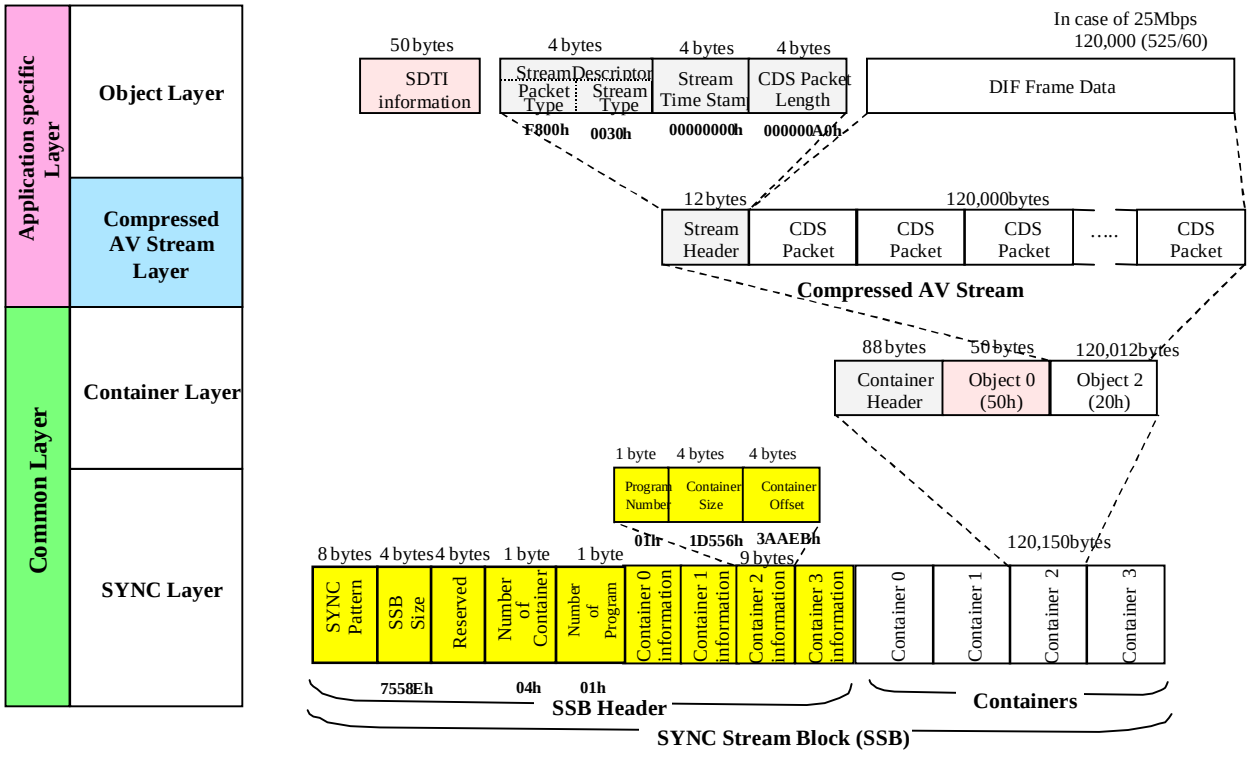


Figure A.4 – Example of the SSB format for four-times-faster-than-real-time transmission of one DV-based audio, data and compressed video

Container offset values are as follows:

{The value of container 0 offset} = 62 bytes
{The value of container 1 offset} = 62 + 120,150 = 120,212 bytes
{The value of container 2 offset} = 62 + (120,150*2) = 240,362 bytes
{The value of container 3 offset} = 62 + (120,150*3) = 360,512 bytes

Other parameter values in figure A.4 are set the same as those in annex A.1.

A.4 SSB format for the combination of two programs of DV-based audio, data, and compressed video, in which one is normal speed transmission and the other is four-times-faster-than-real-time transmission

An example of the SSB format for the combination of two programs of DV-based audio, data, and compressed video, in which one is normal speed transmission and the other is four-times-faster-than-real-time transmission is shown in figure A.5. In figure A.5, there are five containers in one SSB, and each container contains one video frame worth of video, audio, and ancillary data. The SDTI information and the DV-based audio, data, and compressed video of one frame correspond to the data of one video frame worth of video, audio, and ancillary data. In each container, SDTI information is used as object 0, and one frame of DV-based audio, data, and compressed video is used as object 2. In figure A.5, program number 00_h is used for normal speed transmission of a program, and program number 01_h is used for four-times-faster-than-real-time transmission of a program.

In figure A.5, the value of the SSB value length is as follows:

{The value of the SSB value length} = 4 + 1 + 1 + (9 x 5)
+ {The value of container 0 size} + {The value of container 1 size}
+ {The value of container 2 size) + {The value of container 3 size} + {The value of container 4 size} bytes

For 25 Mb/s DV-based audio, data, and compressed video, the value of object size in one frame is 120,012 bytes for 525/60 and 144,012 byes for 625/50. For 50 Mb/s DV-based audio, data, and compressed video data, the value of object size in one frame is 240,012 bytes for 525/60 and 288,012 bytes for 625/50.

For example, for 25 Mb/s DV-based when the FEC word does not exist in the CDS packet (where bit 29 of the stream header = 1), the total length of the CDS packet in one object (object 2) becomes equal to the length of DV-based 1-frame

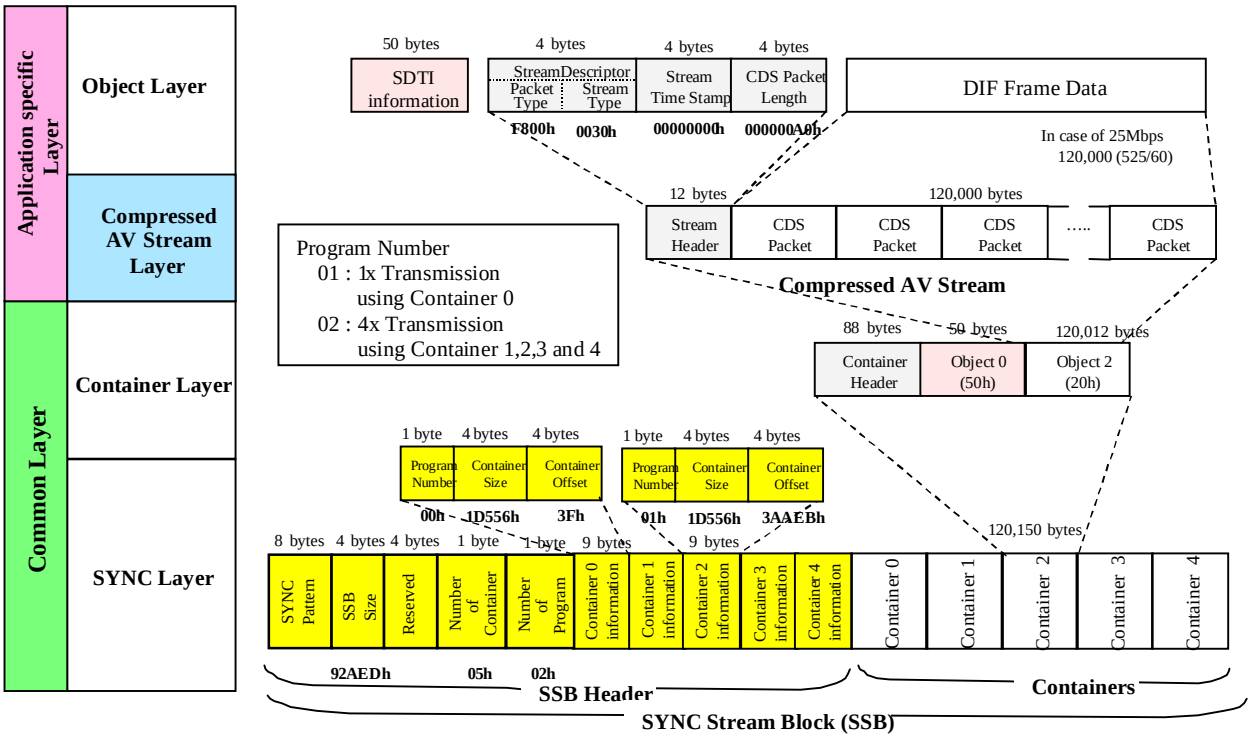


Figure A.5 – Example of the SSB format for the combination of two programs of DV-based audio, data and compressed video, in which one is normal speed transmission and the other is four-times-faster-than-real-time transmission

data. Therefore, the value of container 0 size of 25 Mb/s DV-based 525/60 in one frame is as follows:

{The value of container 0 size for 25 Mb/s DV-based 525/60}
= {The value of container 1 size for 25 Mb/s DV-based 525/60}
= {The value of container 2 size for 25 Mb/s DV-based 525/60}
= {The value of container 3 size for 25 Mb/s DV-based 525/60}
= {The value of container 4 size for 25 Mb/s DV-based 525/60}
= 88 + 50 + 120,012
= 120,150 bytes

The value of the SSB value length of 25 Mb/s DV-based 525/60 in one frame is as follows:

{The value of the SSB value length for 25 Mb/s DV-based 525/60 in one frame}
= 71 + (120,150 x 5)
= 600,821 bytes

Container offset values are as follows:

{The value of container 0 offset} = 71 bytes
{The value of container 1 offset} = 71 + 120,150 = 120,221 bytes
{The value of container 2 offset} = 71 + (120,150*2) = 240,371 bytes
{The value of container 3 offset} = 71 + (120,150*3) = 360,521 bytes
{The value of container 4 offset} = 71 + (120,150*4) = 480,671 bytes

Other parameter values in figure A.5 are set the same as those in annex A.1.

Annex B (informative)
Bibliography

ANSI/SMPTE 125M-1995, Television — Component Video Signal 4:2:2 — Bit-Parallel Digital Interface

ANSI/SMPTE 259M-1997, Television — 10-Bit 4:2:2 Component and 4f_{SC} Composite Digital Signals — Serial Digital Interface

ANSI/SMPTE 267M-1995, Television — Bit-Parallel Digital Interface — Component Video Signal 4:2:2 16x9 Aspect Ratio

ANSI/SMPTE 293M-1996, Television — 720 x 483 Active Line at 59.94-Hz Progressive Scan Production — Digital Representation

ANSI/SMPTE 294M-1997, Television — 720 x 483 Active Line at 59.94-Hz Progressive Scan Production — Bit-Serial Interfaces

SMPTE 274M-1998, Television — 1920 x 1080 Scanning and Analog and Parallel Digital Interfaces for Multiple Picture Rates

SMPTE 292M-1998, Television — Bit-Serial Digital Interface for High-Definition Television Systems

SMPTE 296M-2001, Television — 1280 x 720 Progressive Image Sample Structure — Analog and Digital Representation and Analog Interface

SMPTE 321M-1999, Television — Data Stream Format for the Exchange of DV-Based Audio, Data and Compressed Video over a Serial Data Transport Interface

ITU-T I.356 (03/00), B-ISDN ATM Layer Cell Transfer Performance

ITU-T I.361 (02/99), B-ISDN ATM Layer Specification