SMPTE STANDARD

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

1 Scope

This standard defines a common layer for transport of packetized audio, video, and data over asynchronous transfer mode (ATM) using ATM adaptation layer type 1 (AAL1). This standard covers the transmission of the packetized audio, video, and data including rates at standard time and faster-than-real time, and multiprogram transmission as well. The common layer shall be a higher layer located immediately above the AAL1 layer, and which does not depend on any application. The usage of the transmission packets for particular applications is defined in other documents.

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. The standard is 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.

ANSI/SMPTE 298M-1997, Television — Universal Labels for Unique Identification of Digital Data

SMPTE 336M-2001, Television — Data Encoding Protocol using Key-Length-Value

ISO/IEC 8825-1:1998 (ITU-T X.690), Information Technology — ASN.1 Encoding Rules — Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), Pars. 8.1.3.4 and 8.1.3.5

ITU-T I.363.1 (08/96), B-ISDN ATM Adaptation Layer Specification: Type 1 AAL

3 Common layer

The common layer is the next higher layer located immediately above AAL1 and transports packets of applications. The common layer consists of two sublayers. The lower sublayer is the SYNC layer and the higher sublayer is the container layer.

3.1 SYNC layer

The SYNC layer defines the SYNC stream block (SSB) which includes containers, but does not define the usage of such containers. The container is defined as follows for transmitting packetized audio, video, and data.

3.1.1 SYNC stream block (SSB) format

The SSB data are mapped to AAL1 byte by byte. SSB is transmitted sequentially from the first byte of the SSB header during the time interval of a single video frame. SSB consists of SSB header and one or more containers. The SSB format is shown in figure 1 and shall include the following:

UL key: 16 bytes
- UL header: 2 bytes
- UL designators: 6 bytes
- Item designator: 8 bytes

SSB value length: Variable (defined in 3.1.3)

SSB value: Variable
- Reserved: 4 bytes
- Number of containers: 1 byte
- Number of programs: 1 byte
- Container information blocks: 9 bytes
- Containers: The length of each container is indicated in each container information block.

Approved
January 31, 2002

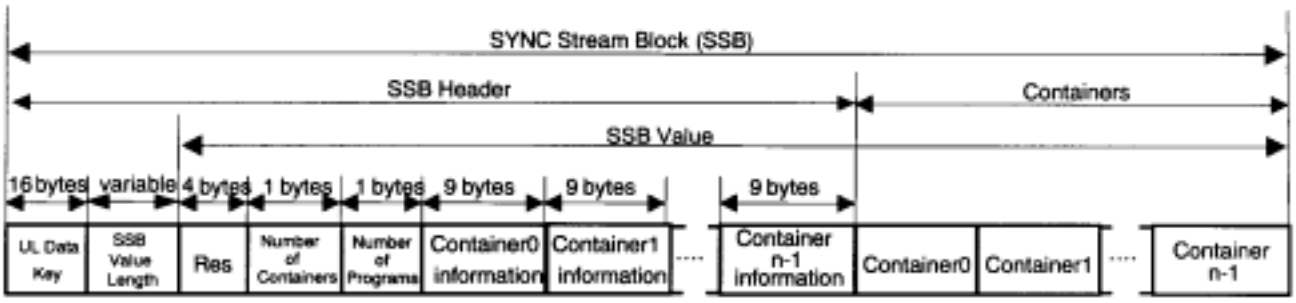

Figure 1 – SYNC stream block (SSB) format

3.1.2 UL key

A universal label (UL) identification key (UL key) is composed of a UL header, one element of UL designators, and an item designator.

3.1.2.1 UL header

The UL header shall be positioned at the beginning of the SSB. The UL header consists of 2 bytes that are set to 0x06 and 0x0E.

3.1.2.2 UL designators

The UL designators shall be positioned immediately after the UL header. The UL designators consist of 6 bytes that are 0x2B, 0x34, 0x03, 0x01, 0xxx, and 0x01.

3.1.2.3 Item designator

The item designator bytes shall be positioned immediately after the UL designators. Two types of item designator patterns are defined, item designator-0 and item designator-1. Item designator-0 consists of 8 bytes that are 0xE3, 0xCB, 0xAA, 0x4C, 0xEA, 0xCD, 0x7A, and 0x81. Item designator-1 consists of 8 bytes that are 0xE3, 0xCB, 0xAA, 0xCD, 0x7A, 0xEA, 0x81, and 0x4C. Each item designator pattern consists of 8 bytes. The first 3 bytes, bytes 0, 1, and 2, of item designator-0 have the same values as bytes 0, 1, and 2 of item designator-1 respectively.

Item designator-0 and item designator-1 are used alternately in the SSB sequence at the AAL service access point (AAL-SAP). The item designator pattern is a reference of each video frame; therefore, the video frame rate of each container within one SSB shall be the same.

3.1.3 SSB value length

The SSB value length indicates the length in bytes of the distance from the first byte of the reserved data field to the last byte of the last container. The SSB value length field shall be encoded using the basic encoding rules (BER) for either the short-form or the long-form encoding of length octets specified in ISO/IEC 8825-1, paragraph 8.1.3. The ATM common layer transport limits the length of the SSB value to 6 octets.

3.1.4 SSB value

The SSB value is composed of reserved data, number of containers, number of programs, container information blocks, and containers.

3.1.4.1 Reserved data space

The reserved data space shall consist of 4 bytes.

3.1.4.2 Number of containers

The number of containers shall consist of 1 byte. It indicates the number of containers (n) within the SSB.

3.1.4.3 Number of programs

The number of programs shall consist of 1 byte. It indicates the number of programs within the SSB.

3.1.4.4 Container information block

The container information block shall consist of 9 bytes. The number of container information blocks is the same number as the number of containers (n) within the SSB. Each container information block is assigned a number from 0 to (n - 1), indicating to which container in the container sequence within the SSB the information belongs. The first container information block is assigned 0. Each container information block consists of the program number, the container size, and the container offset. The container information block mapping is shown in figure 2

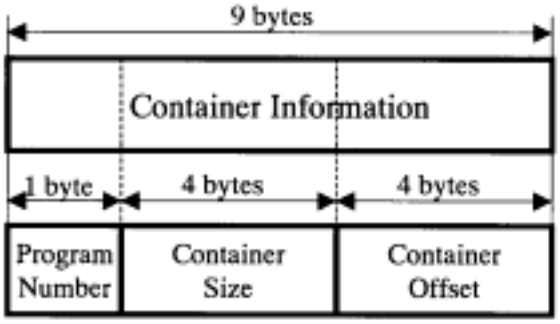

Figure 2 – Container information

3.1.4.4.1 Program number

The program number shall consist of 1 byte. It indicates the program number of the container data within the SSB. It is assigned sequentially from 0 to total program number – 1.

3.1.4.4.2 Container size

The container size shall consist of 4 bytes. It indicates the size of the container in bytes.

3.1.4.4.3 Container offset

The container offset shall consist of 4 bytes. It indicates the header byte position of the container from the start byte of the SYNC stream block.

3.1.4.5 Container

The container is designed for transporting packetized data. The length of the container is indicated by container information described in 3.1.4.4. The container does not contain its SSB sequence number, but it is assigned a different 4-byte sequence number as an identifier.

3.2 Container layer

The container layer defines the structure of the container in 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: simple mode and extended header mode. The simple mode supports four objects of somewhat restricted types. The extended header mode, needed when there are more than four objects, supports up to 16 objects. The container format is shown in figure 3. The container with m objects shall include the following:

Container header: 88 bytes

Extended container header (if needed): (m-4)*4 bytes (m = > 4)

Object: The length is indicated in the container header or extended container header.

3.2.1.1 Container header

The container header shall consist of 88 bytes and each word consists of 4 bytes. The first 6 words of the header apply to the entire container. The remaining 16 words are divided into four identically structured sections, one for each data object. The container header shall include the following and is shown in figure 4:

Sequence number: 4 bytes
Clip ID: 4 bytes
Container time stamp: 8 bytes
Video frame rate: 1 byte
Transmission rate: 1 byte
Reserved: 2 bytes
Mode: 1 byte
Number of objects: 1 byte
Reserved: 1 byte
Size of extended header: 1 byte
Object 0 information: 4 words (16 bytes)
Object 1 information: 4 words (16 bytes)
Object 2 information: 4 words (16 bytes)
Object 3 information: 4 words (16 bytes)
(Each object information has class, size, offset, and object type defined.)

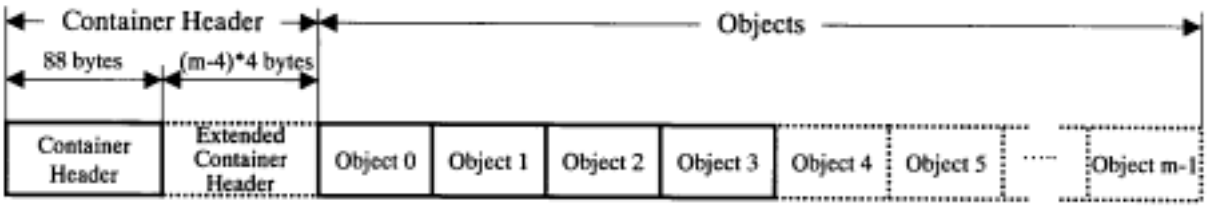

Figure 3 – Container format

Word	Identifier	BYTE 0	BYTE 1	BYTE 2	BYTE 3
0	Sequence number	MSB	-number-	-number-	LSB
1	Clip ID	MSB	-clip ID-	-clip ID-	LSB
2	Container time stamp	MSB	-time stamp-	-time stamp-	-time stamp-
3	Container time stamp	-time stamp-	-time stamp-	-time stamp-	LSB
4	Transmission type	Video frame rate	Transmission rate	Reserved	Reserved
5	Container type	Mode	Number of objects	Reserved	Size of extended header
6	Object 0 class	Type $5x_h$	Link pointer	Index (xx_h)	Index (xx_h)
7	Object 0 size	MSB	-size-	-size-	LSB
8	Object 0 offset	MSB	-offset-	-offset-	LSB
9	Object 0 Object type defined	Object type defined	Object type defined	Object type defined	Object type defined
10	Object 1 class	Type $4x_h$	Link pointer	Index (xx_h)	Index (xx_h)
11	Object 1 size	MSB	-size-	-size-	LSB
12	Object 1 offset	MSB	-offset-	-offset-	LSB
13	Object 1 Object type defined	Object type defined	Object type defined	Object type defined	Object type defined
14	Object 2 class	Type $1x_h$ or $2x_h$	Link pointer	Index (xx_h)	Index (xx_h)
15	Object 2 size	MSB	-size-	-size-	LSB
16	Object 2 offset	MSB	-offset-	-offset-	LSB
17	Object 2 Object type defined	Object type defined	Object type defined	Object type defined	Object type defined
18	Object 3 class	Type $1x_h$ or $2x_h$	Link pointer	Index (xx_h)	Index (xx_h)
19	Object 3 size	MSB	-size-	-size-	LSB
20	Object 3 offset	MSB	-offset-	-offset-	LSB
21	Object 3 Object type defined	Object type defined	Object type defined	Object type defined	Object type defined

Figure 4 – Container header

3.2.1.1.1 Sequence number

A 32-bit unsigned integer used to identify where, in a series of containers representing a video clip, this objects fits.

3.2.1.1.2 Clip ID

A 32-bit unsigned integer used for identification of a video clip. Intended for use by a higher level protocol as a way to identify the container as an element of a specific video clip.

3.2.1.1.3 Container time stamp

A 64-bit word used to store a time stamp for transmission purposes. This time stamp represents the container time. There may be other time stamps and time codes associated with each object.

The time stamp format is described as the ANSI X3.230 (FC-PH) expiration time which is a 64-bit fixed-point number valued in seconds. The integer part of the number is the most significant 32 bits and the fractional part is the least significant 32 bits. The maximum integer is therefore 2^{32} - 1, and there are 32 bits for the fraction of a second. The timer start time is 1 January 1900, 0 hours UTC. The timer rolls over every 136 years (4,294,967,295 seconds), but this can be easily detected in hardware or software.

3.2.1.1.4 Video frame rate

The encoding of the video frame rate field shall be as shown in table 1.

Table 1 – Video frame rate

Frame rate per second	Code
Null	00_h
15	01_h
20	02_h
24	03_h
24*1000/1001	83_h
24 (Segmented frames)	23_h
24*1000/1001 (Segmented frames)	$A3_h$
25 (PAL)	44_h
30	45_h
30*1000/1001 (29.97 NTSC)	$C5_h$
50	06_h
60	07_h
60*1000/1001 (59.94)	87_h

Bit 7 is used to indicate the presence of the factor 1000/1001 in the frame rate (as with NTSC).

Bit 6 is used to indicate interlacing.

Bit 5 is used to indicate progressively scanned but segmented frames.

3.2.1.1.5 Transmission rate

The transmission rate is an 8-bit signed integer. A positive value of n represents a transmission speed of (n/4)*(the video frame rate). A negative value of n represents a transmission speed of (-4/n)*(the video frame rate).

NOTE – For example, the transmission speed will be four times the frame rate for n = 16, and the transmission speed will be 1.25 or 1.5 times the frame rate for n = 5 or 6. If the transmission speed is two-thirds of the frame rate, n would be (–6).

3.2.1.1.6 Mode

Simple or extended mode:

- Bit 7 = 0, Simple mode
- Bit 7 = 1, Extended header mode

3.2.1.1.7 Number of objects

The integer number of objects in the container.

3.2.1.1.8 Size of extended header

In extended header mode, with more than four objects, the size is equal to the total number of objects minus 4, multiplied by 16 bytes. The size can be positive or 0 (no extended header) only.

3.2.1.1.9 Object descriptions

Each object has a 4-word field in the header to describe relevant information about that object. The generic object shall include the following:

Object n type: 1 byte
Link pointer: 1 byte
Object n index: 2 bytes
Object n size: 4 bytes
Object n offset: 4 bytes
Object n object type defined: 4 bytes

3.2.1.1.9.1 Object n type

See 3.2.1.3.

3.2.1.1.9.2 Link pointer

An unsigned byte used to link an object with another object within the same container. The value is the object number of the associated object.

NOTE – When there are k pieces of objects (object n1, object n2, ..., object nk, where n1 < n2 < ... < nk) which have relationships to each other, then the value of link pointer in object n1 is the second smallest object number n2, and the values of link pointers in the other objects (object n2, ... object nk) are the smallest object number n1.

3.2.1.1.9.3 Object n index

See 3.2.1.3.

3.2.1.1.9.4 Object n size

Object size in bytes.

3.2.1.1.9.5 Object n offset

The object offset in bytes from the beginning of the container.

3.2.1.1.9.6 Object n object type defined

The use of this field is defined by the object type.

3.2.1.2 Extended container header

An extended header is defined to address requirements of applications not directly supported by the simple mode. Provisions include the support of a larger and variable number of objects. The extended header is a logical extension of the container header. It is physically located directly after the container header and before object 0. Object descriptions follow the same format as the object descriptions within the container header. There are no restrictions as to the object type for any of the objects addressed by the extended header. The length of the extended header is dependent on the number of objects in the container. The extended header length equals 4 words for each object beyond object 3. The maximum number of objects is 16, thus the maximum length is 4 words * (16-4) = 48 words. The extended header is shown in figure 5.

3.2.1.3 Object classification system

The container system provides a mechanism for defining the bit and byte packing of audio-video data

Word	Identifier	BYTE 0	BYTE 1	BYTE 2	BYTE 3
22	Object 4 class	Type xx_h	Link pointer	Index (xx_h)	Index (xx_h)
23	Object 4 size	MSB	-size-	-size-	LSB
24	Object 4 offset	MSB	-offset-	-offset-	LSB
25	Object 4 Object type defined	Object type defined	Object type defined	Object type defined	Object type defined
26	Object 5 class	Type xx_h	Link pointer	Index (xx_h)	Index (xx_h)
27	Object 5 size	MSB	-size-	-size-	LSB
28	Object 5 offset	MSB	-offset-	-offset-	LSB
29	Object 5 Object type defined	Object type defined	Object type defined	Object type defined	Object type defined
.....					
(m-1)*4 +6	Object m-1 class	Type xx_h	Link pointer	Index (xx_h)	Index (xx_h)
(m-1)*4 +7	Object m-1 size	MSB	-size-	-size-	LSB
(m-1)*4 +8	Object m-1 offset	MSB	-offset-	-offset-	LSB
(m-1)*4 +9	Object m-1 Object type defined	Object type defined	Object type defined	Object type defined	Object type defined

Figure 5 – Extended container header

types. It does not define audio or video formats, but rather the packing of standard formats into objects for efficient transport by ATM.

The object classification is represented in the header by the object class word. This word is composed of two key elements, the type byte (1 byte) and index bytes (2 bytes). The type byte is a coarse classification of the data type. Each value of type byte represents a table of data types. The index bytes key into the table.

In addition to the index values assigned in this standard, there are three reserved classifications at both the type and index levels:

a) Reserved – for future standard assignment;

b) Negotiated – for use by a higher level protocol to dynamically assign data types;

c) Vendor specific – for open use and guaranteed not to be assigned in the future.

This standard makes no guarantee of interoperability when using vendor specific types of indices.

The informative object table hierarchy is shown in figure 6.

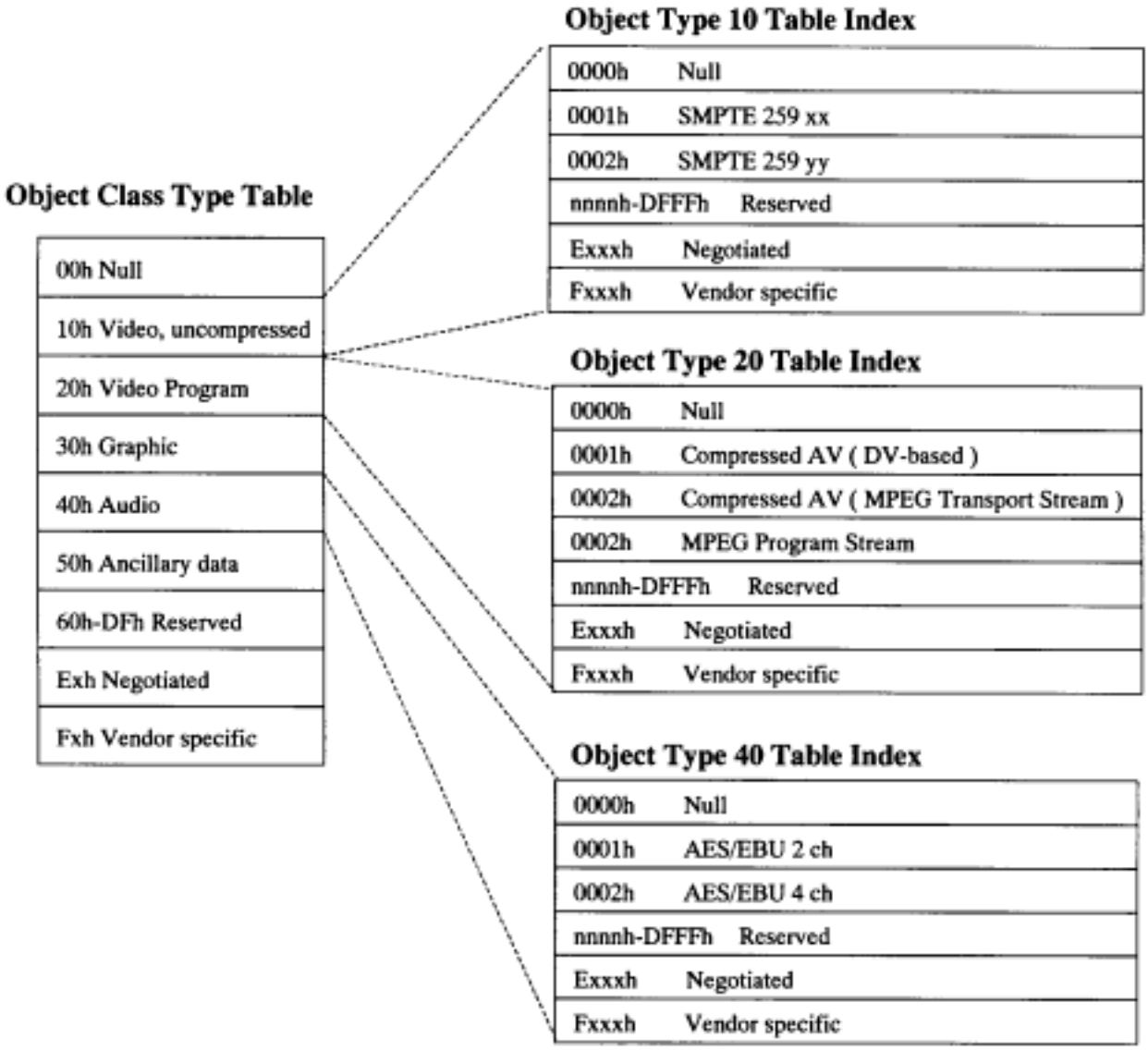

Figure 6 – Object table hierarchy

3.2.1.3.1 Object type bytes

Object type bytes are unsigned bytes. The value represents a table. The values assigned are listed in table 2. Values listed as reserved are for future assignment by ANSI. Values labeled vendor specific are intended for proprietary use by a specific vendor and will not be reassigned in the future. Note that in each table pointed to by the type byte, there are also vendor specific values. The intent is to allow a vendor to use an assigned type classification (such as uncompressed video) with a proprietary format. Therefore, vendor specific type bytes should only be used for types not listed in table 2. Undefined object type byte numbers are reserved.

Table 2 – Object class type

Type	Code
Null object	00_h
Video, uncompressed	10_h
Video, compressed	11_h
Reserved for future video	12_h to $1F_h$
Video program	20_h
Reserved for video program (multiplexed stream)	21_h to $2F_h$
Graphics	$3x_h$
Audio, uncompressed	41_h
Audio, compressed	42_h
Ancillary data	$5x_h$
Reserved	60_h to DF_h
Negotiated object types	$E0_h$ to EF_h
Vendor specific	$F0_h$ to FF_h

3.2.1.3.2 Object index bytes

The object index bytes are defined to be a 2-byte unsigned value representing an index to the table indicated by the object type byte of the same object class word. For the case of a null object, the bytes are set to 0000_h.

4 Transmission order

4.1 Faster-than-real-time transmission

In the case of faster-than-real-time transmission (n times transmission), n pieces of container are used within a single SSB. One video frame worth of video, audio, and ancillary data is mapped into one container in time sequence. N pieces of container are transmitted continuously. An example is shown in figure 7.

4.2 Multiprogram transmission

In the case of multiprogram transmission, one video frame worth of video, audio, and ancillary data is mapped into one container and all programs are multiplexed into SSB. The program number shall be indicated by program number in the container information. The order of the program is the same within every SSB. An example is shown in figure 8 (p program transmission).

4.3 Combination of faster-than-real-time transmission and multiprogram transmission

In the case of a combination of faster-than-real-time transmission and multiprogram transmission, all programs are multiplexed into SSB. The program number shall be indicated by program number in the container information. Plural containers of faster-than-real-time programs are transmitted continuously and the order of the program is the same within every SSB. An example in the case of the 3-program transmission (program 0: 1 time; program 1: 4 times; program 2: 1 time) is shown in figure 9.

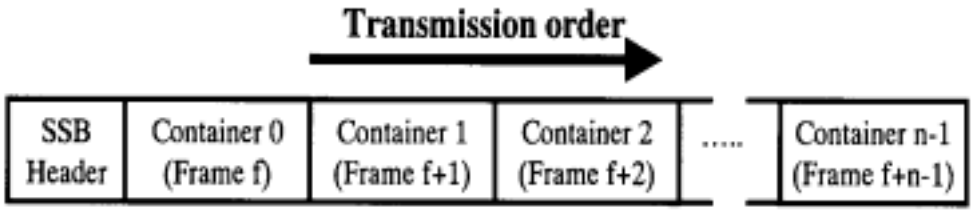

Figure 7 – Faster-than-real-time transmission

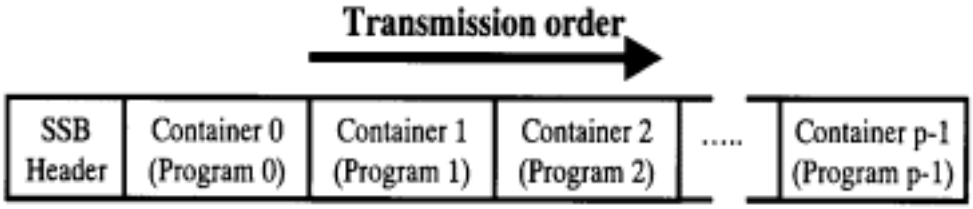

Figure 8 – Multiprogram transmission

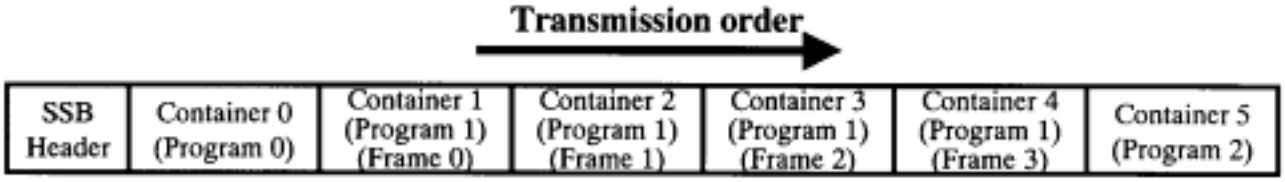

Figure 9 – Combination of faster-than-real-time transmission and multiprogram transmission

Annex A (informative)
SSB sequence

As described in 3.1.2.3, item designator-0 and item designator-1 are used alternately in the SSB sequence. The overview of the SSB sequence is shown in figure A.1.

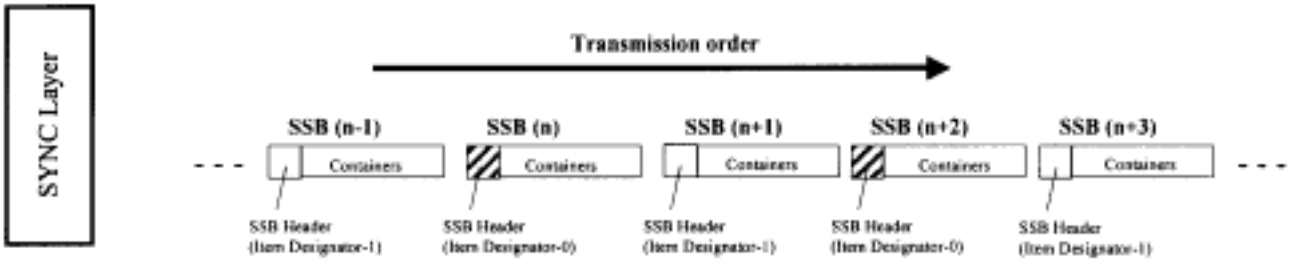

Figure A.1 – SSB sequence

Annex B (informative)
Bibliography

ANSI X3.230-1994 (R1999), Information Technology — Fibre Channel — Physical and Signaling Interface (FC-PH)

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 16×9 Aspect Ratio

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

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

ANSI/SMPTE 296M-1997, Television — 1280 × 720 Scanning, Analog and Digital Representation and Analog Interface

SMPTE 274M-1998, Television — 1920 × 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

ISO/IEC 8824-1:1998 (ITU-T X.680), Information Technology — Abstract Syntax Notation One (ASN.1) — Specification of Basic Notation

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

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