

for Television —

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

Page 1 of 10 pages

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.

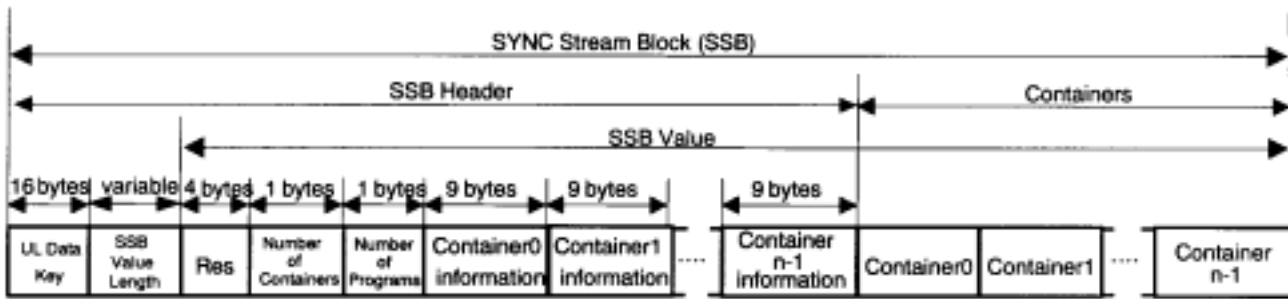


Figure 1 – SYNC stream block (SSB) format

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

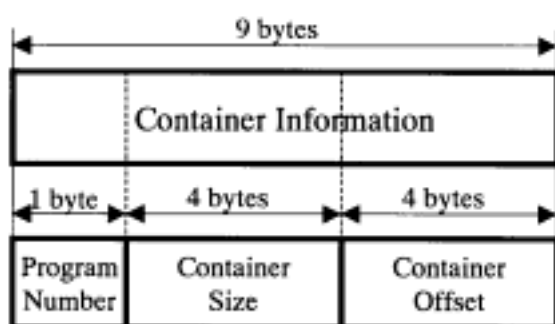


Figure 2 – Container information

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.)