RECOMMENDATION ITU-R BS.776*

Format for user data channel of the digital audio interface**

(1992)

The ITU Radiocommunication Assembly

considering

a) that there is a real need for a flexible format, independent of any specific application, for user data channel of the digital audio interface (see Recommendation ITU-R BS.647);

b) that there are advantages if all equipment use the same format of transmission to exchange ancillary data;

c) that the format should be able to carry efficiently a wide variety of information, short and long messages, related in time to the audio data or not;

d) that the HDLC frame (high level data link control: ISO/IEC 13239 (2000)) is a widely used standard in the information technology industry,

recommends

that the method of formatting user data channel, together with the rules for data insertion into the multiplex and for data management described in Annex 1 should be adopted.

Annex 1

Format for user data channel of the digital audio interface

1 Introduction

This annex is intended to be a companion document to the digital audio interface specification (see Recommendation ITU-R BS.647). It was prepared as a result of a desire by users of the interface to have a recommended format for the user data channel provided by the user bit.

The requirement was for a system which is flexible, and independent of the user application, and is able to carry message data related in time to the audio data as well as information such as text which may be unrelated to the audio. A further requirement was for a constant data rate within a range of ± 12.5% of a sampling frequency of 48 kHz. The system specified is based on a widely used packet communication protocol, high level data link control, HDLC (ISO/IEC 13239 (2000)) which is standardized in the information technology industry. The HDLC protocol has been adapted for unidirectional transmission and to permit the accurate transmission of time-dependent

---

* Radiocommunication Study Group 6 made editorial amendments to this Recommendation in 2003 in acordance with Resolution ITU-R 44.

** This Recommendation should be brought to the attention of the IEC, the Audio Engineering Society (AES) and the SMPTE.

information. However, the readily available HDLC integrated circuits which are fabricated by several manufacturers are still able to be used. It is therefore expected that this transmission format will be easy to implement without recourse to special hardware and will, as a result, be included as a matter of routine in commercial interface equipment.

This specification of the transport system is in two basic parts. The first covers the data formatting and defines the mechanisms which transform the messages into packets to allow the multiplexing of simultaneous messages into the network. The second part describes the channel management and defines the use of the channel and the multiplexing rules.

The characteristics of the system follow closely the general principles of the ISO layer model.

2 Scope

This annex specifies a recommended method of formatting the user bit of each channel of the digital audio interface (Recommendation ITU-R BS.647). Each user data channel so formed is independent of the user application and is primarily intended for the transmission of data associated with the audio signal although data unrelated to the audio may also be transmitted. There is no restriction on the length of messages which may be transmitted in the user data channel. The data capacity is constant for sampling frequencies in the range 42 kHz to 54 kHz, i.e. 48 kHz ± 12.5%. Sampling frequencies outside this range can be accommodated, but if sample rate conversion to a lower sampling frequency is performed, data management is to be employed to avoid loss of vital data.

The annex describes the method of formatting user information into packets together with the rules for data insertion into the multiplex and for data management. A separate Recommendation is in preparation which recommends a strategy for hardware and software addressing. The purpose and content of the user information for particular applications are outside the scope of this annex.

This annex describes a practice for professional applications and is not intended to encompass applications related to consumer versions of the digital audio interface.

3 General

3.1 The digital audio interface is specified in Recommendation ITU-R BS.647 and is primarily intended to allow the transmission of audio signals between digital audio equipment. This annex specifies a system by which the user data channel can be used to transport a wide variety of messages along with the audio data. The messages can come from many different applications, scripts, sub-titles, editing information, copyright, performers credits, downstream switching instructions, etc. The application can be freely chosen and defined by the user and the messages which are sent can be of many different types.

3.2 The transport system is based on a widely used protocol, high level data link control, HDLC, (ISO/IEC 13239 (2000)) which is standardized in the information technology industry. In general the HDLC system is capable of handling messages in two directions, but in this system it is only used in one, the same direction as the audio signals. Integrated circuits are available from several manufacturers which support the HDLC protocol. These often take the form of the HDLC interface combined with a microprocessor and a local interface to the message receiving or sending system. Thus it is hoped that the transport system will be easy to implement without special hardware and be included as a routine matter in commercial interface equipment.

3.3 Digital audio signals are likely to be repeatedly processed and passed on between different equipment in broadcasting studios. The messages which the user would like to accompany the audio signals are likely to grow at successive stages. This transport system allows downstream equipment to add or remove messages in the channel, provided there is enough capacity. Each message is sent as one or more packets, depending on its length. Each packet carries the address of its destination. This allows a receiver to read only those messages addressed to it. A large number of addresses are available and the system puts no restriction on the user as to how the addresses are used or for what purpose the messages are used.

3.4 Main features

The main features of the transport system from the point of view of the user are as follows.

3.4.1 Messages

The transport system can be used to carry a wide variety of information. Messages can be of any length. Messages can be time critical or not.

3.4.2 Multiplexing

The transport system can carry simultaneous messages from different applications, subject to the maximum bit rate provided by the interface. The system allows the user to insert messages at any point of the chain. The messages can have different priority levels, which effects how quickly and how often the parts of a message can be inserted.

3.4.3 Bit rate

The transport system is based on packets and can be used with any audio sample rate. For audio sample rates within the range of 48 kHz ± 12.5%, which includes 44.1 kHz, it provides a constant message information rate. This is achieved by inserting dummy or justification bits at higher sample frequencies.

3.4.4 Synchronization

The transport system allows the user data channel to become a self-contained transmission channel which is independent of the audio signal block structure. However the channel can easily be synchronized to an external signal such as a time code or a video frame rate if this is required by the user.

3.4.5 Error detection

The transport system includes error checking which allows the detection of any corruption of the data in the messages. The user can safeguard important messages by instructing the transport system to automatically repeat every packet, or by repeating the message itself.

3.4.6 Efficiency

The system has an efficiency given by the ratio of the application data bit rate to the total bit rate of the user data channel.

The system needs a certain amount of overhead bit rate to provide addressing, packet identification, error detection, justification, etc. The percentage amount of this overhead varies predominantly with the length of the message and to a lesser extent to block length.

The efficiency can be as high as 60% with a block length of 40 ms and sample frequency of 48 kHz. 70% efficiency is achievable with the same block length and a sample frequency of 44.1 kHz because fewer justification bits are required.

4 Terminology

4.1 Transport system

The method by which messages are carried between the source and destination of the application.

4.2 Address

The identification of either the destination hardware or the application. It can be combined with an address extension.

4.3 Address – Extension

The part of the address which extends the range destination hardware or applications which may be addressed.

4.4 Block

The repetitive structure chosen for the transport system by the user.

4.5 Continuity index

A count of the messages or packets sent to a destination. In this system the count is modulo 8, allowing up to seven missing messages or packets to be detected. This system uses a message continuity index and a packet continuity index.

4.6 Control byte

Contains information enabling receiving equipment to interpret and decode successfully and reliably the data which follows. The control byte has four fields for a priority index, a packet continuity index, address extension enable and link bits.

4.7 Cyclic redundancy check code (CRCC)

See frame check sequence.

4.8 Frame check sequence (FCS)

Two bytes formed by a mathematical manipulation of the message information that are added to the packet to provide forward error detection.

4.9 High level data link control (HDLC)

An internationally standardized protocol for the transmission of messages based on packets.

4.10 Frame level

The level of the transport system where the packets are formed into HDLC frames.

4.11 HDLC frame

The HDLC packet after it has been channel coded, with the addition of the frame check sequence and the limit flags (ISO/IEC 13239 (2000).

4.12 HDLC frame flag

A unique code at the physical level which identifies the start and end of an HDLC frame.

4.13 Justification bits

Bits which are left unused in the user data channel to allow the system to operate between audio sample frequencies 42 kHz and 54 kHz without losing information. The number of justification bits varies with the audio sample frequency.

4.14 Message level

The level of the transport system where the messages are received from the application and processed so that they can be formed into packets. At the receiver packets are processed, reformed into messages and passed to the application.

4.15 A_0. The least significant bit (LSB) of a byte

Bit transmitted first in this serial transmission system.

4.16 A_7. The most significant bit (MSB) of a byte

Bit transmitted last in this serial system.

4.17 Packet level

The level of the transport system where the self-contained packets which compose the message are formed.

4.18 Physical level

The level of the transport system where the frames are channel coded and inserted in the user data channel of the digital audio interface.

4.19 Priority index

The number chosen by the user by which a priority is assigned to a packet or message. Four priority levels have been provided.

4.20 Repetition index

The number chosen by the user which controls the number of times each packet of a message is repeated.

4.21 Segment

A part of a long message after it has been split up. The segment will form the contents of a packet.

4.22 Message delimiters

The unique indicators of the start and finish at the message.

5 Data formatting

5.1 The application level

The messages are generated by the application, and are then passed to the transport system together with the other parameters which the transport system uses to process the messages. The parameters are:

– the destination address;

– a priority level;

– message delimiters;

– a repetition index.

5.1.1 Destination address

The application shall provide a destination address. It is either 1 byte long, or 2 bytes if an address extension is used.

5.1.2 Priority index

The priority index shall be chosen by the user according to the urgency of the message. It is one of four possible levels, and determines either the maximum delay before the message is sent or how quickly it is sent if it is a long message. The priority system is described in more detail in § 6.3.2.

5.1.3 Message delimiters

The application level must provide two flags which indicate the beginning and the end of the message.

5.1.4 Repetition index

The repetition index shall control how many times each part of the message shall be repeated by the transport system. The repetition index can have any value, but to minimize the system loading, a repetition index between 0 and 5 is recommended. The whole message can also be repeated by the application.

5.1.5 Message

The message content shall be treated as a binary signal and it can be of any number of 8-bit bytes.

5.2 The transport system

The transport system shall receive the message data from the application level and processes it so that it can be transmitted in the user data channel. The data formatting shall be carried out on four levels (Fig. 1):

– the message level;

– the packet level;

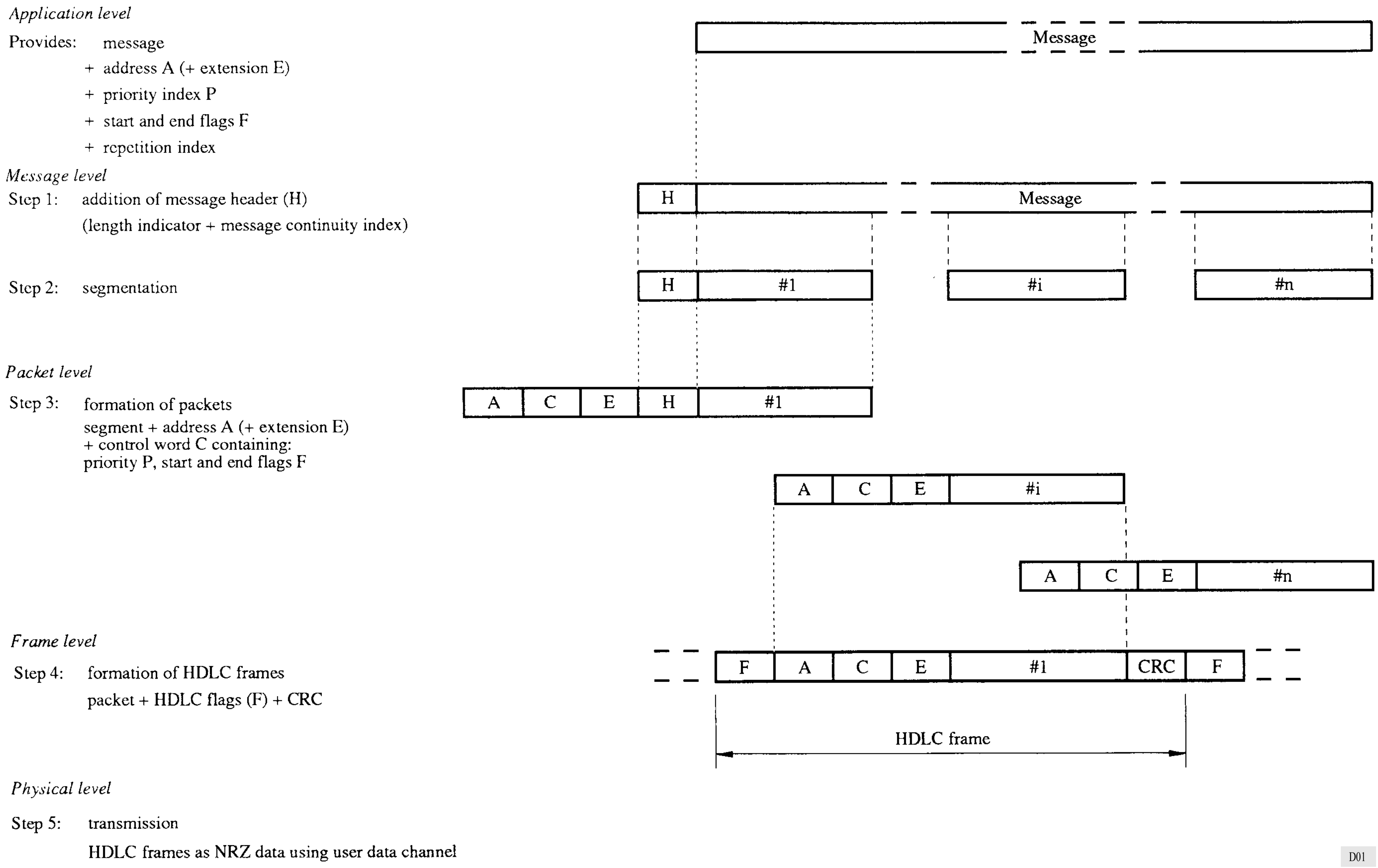
– the frame level;

– the physical level.

Each level has its own function and all the levels together ensure the correct formatting of the data to allow it to be transported.

FIGURE 1

Main steps in data formatting



5.2.1 The message level

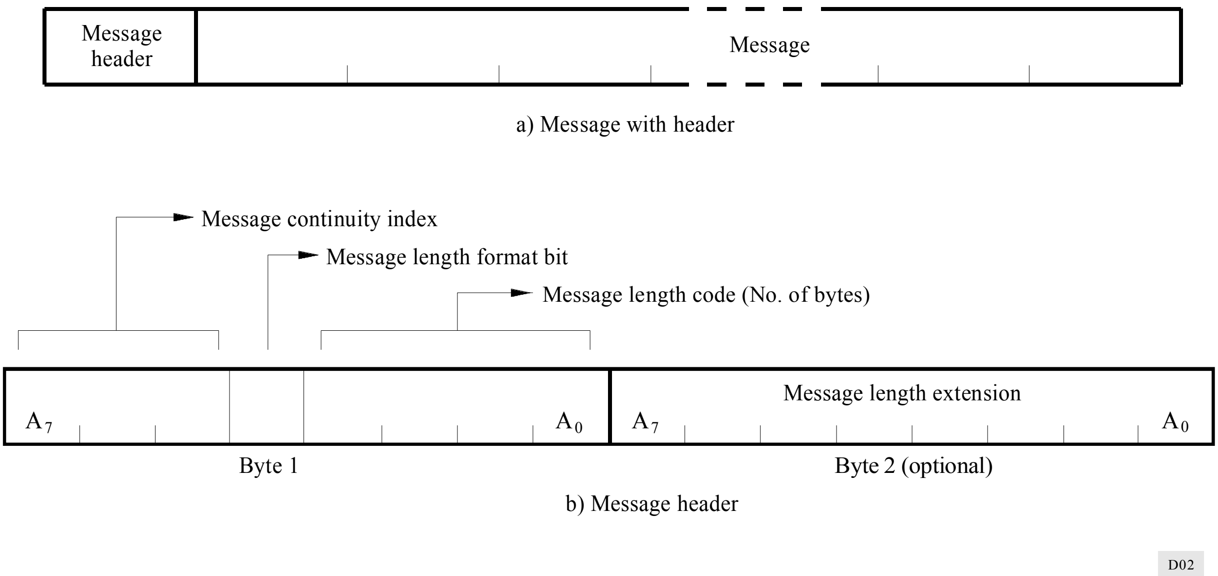
The message level (Fig. 1 – Step 1) shall receive the message from the application together with the other parameters listed in § 5.1. It shall add a header to the message and then divide the message, including the header, into segments which will be transported separately. Finally the message level shall pass on the segments, together with the parameters, to the packet level.

5.2.1.1 Message header

The message header (Fig. 2) shall contain information on the message length and a message continuity index. The header is either one or two bytes long, depending on the length of the message. A message length format bit in the first byte shall signal whether the second byte, the message length extension, is used.

FIGURE 2

Message header



Message header: first byte (A_7, MSB to A_0, LSB)

bits A_7, A_6, A_5 Message continuity index

The three bit continuity index shall be a modulo-8 count of the messages sent by an application. A_7 is the MSB. The receiver can use this index to detect any loss of messages. Each application shall have its own continuity index.

However, when a message is repeated under the control of the repetition index the message continuity index shall not change (see § 3.2).

bit A_4 Message length format

0 The header is one byte long

1 The message length extension byte is used and the header is 2 bytes long.

bits A_3, A_2, A_1, A_0 Message length code most significant bits

These bits indicate the length of the message. The precise use is described in § 5.2.1.2.

Message header: second byte (A_7, MSB to A_0, LSB) (optional)

bits A_7 to A_0 Message length extension code

These bits allow the length code of the message to be extended as described in § 5.2.1.2.

5.2.1.2 Message length coding

The messages can be any length and the different message lengths shall be coded in one of three ways according to the length. (Note that the message length does NOT include the message header byte or bytes.)

– If the length of the message is less than or equal to 15 bytes, the message header shall be one byte long since the message length can be given by the four bit message length code (header byte 1, bits A_3 to A_0). The message length format bit (header byte 1, bit A_4) is set to logic "0".

– If the length of the message is greater than 15 bytes but less than or equal to 4 094 bytes, the address extension byte shall be used and the message length shall be coded by 12 bits (header byte 1, bits A_3 to A_0 plus the extension byte (header byte 2, bits A_7 to A_0)). The message length format bit of the first byte is set to logic "1".

– If the message length is greater than 4 094 bytes or unknown, the message length shall be coded as 4 095 (FFF hex). The message delimiters supplied by the application shall be used to indicate the limits of the message (see § 5.2.2.1).

5.2.1.3 Segmentation

The message, complete with its header, shall be divided into segments of 16 bytes or less if it is the last segment or only segment (Fig. 1 – Step 2). The header shall always be part of the first segment.

5.2.2 Packet level

The packet level shall assemble the segments into packets (Fig. 1 – Step 3). From the message level it shall receive:

– the segments of the message;

– the parameters which were passed on from the application:

  – the address (and the address extension if there is one);

  – the priority index;

  – the beginning and the end delimiters of the message;

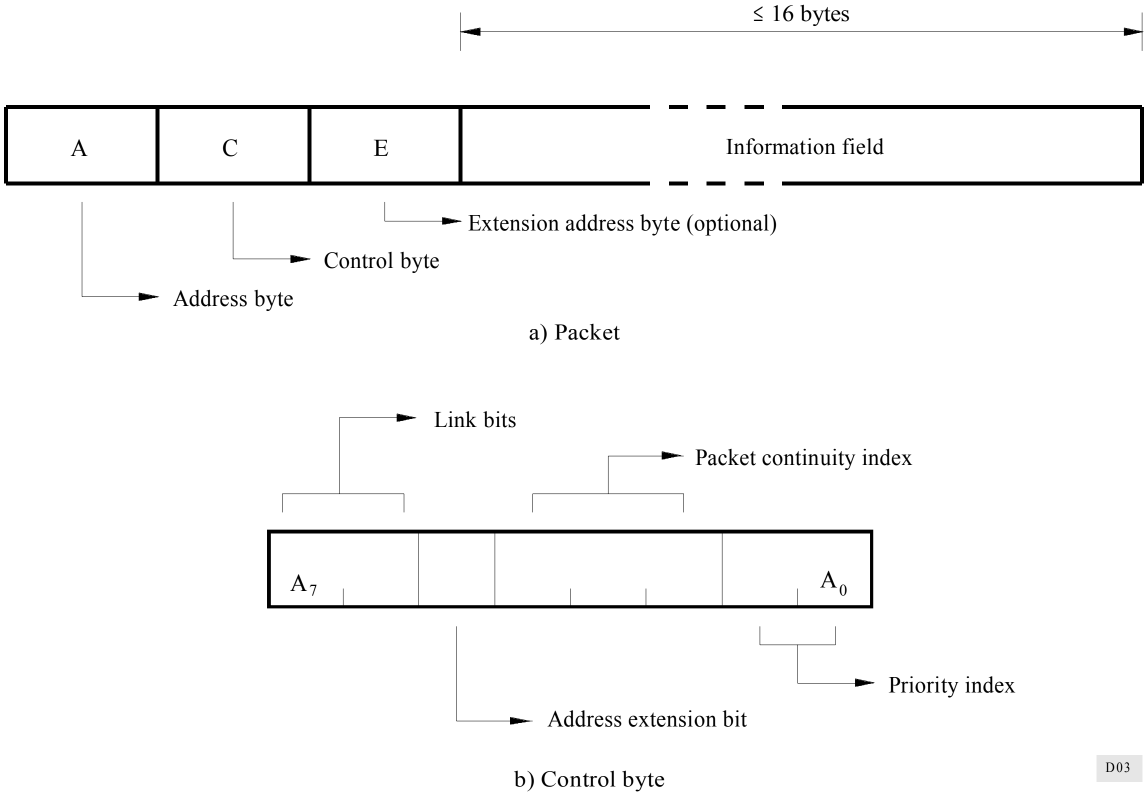
  – the repetition index.

Each packet (Fig. 3a)) shall be formed from one segment of the message with the addition of an address byte, a control byte and an address extension byte if present. The segment shall become the information field of the packet.

The segment, the address byte and the address extension byte, shall be inserted into the packet exactly as they are received from the message level.

FIGURE 3

Packet level



5.2.2.1 Control byte

The control byte shall be generated at the packet level and shall be 8 bits long (A_7, MSB to A_0, LSB), divided into four fields (Fig. 3b)):

bits A_7 A_6 Link bits

1 0 The packet is the first or only packet of the message

(This packet always contains the message header)

0 1 The packet is the last packet of a message which is two or more packets long

0 0 The packet is an intermediate packet of a message which is three or more packets long

1 1 Reserved for system packet identification (see § 6.2.1)

bit A_5 Address extension bit

0 There is no address extension byte

1 There is an address extension byte

bits $A_4\ A_3\ A_2$ Packet continuity index

The packet continuity index shall be a modulo-8 count of all the packets which are sent with the same address defined by an application. A_4 shall be the MSB. This allows the receiver to detect if a packet of a message is missing.

When a packet is successively repeated, under the control of the repetition index, the packet continuity index shall not change.

bits $A_1\ A_0$ Priority index

0 0 Priority index 0 (lowest)

0 1 Priority index 1

1 0 Priority index 2

1 1 Priority index 3 (highest)

NOTE 1 – The priority index is set by the application (see § 5.1.2).

5.2.2.2 Packet level repetition

After the packets are formed, they shall be repeated as many times as defined by the repetition index and passed on to the frame level.

5.2.3 Frame level

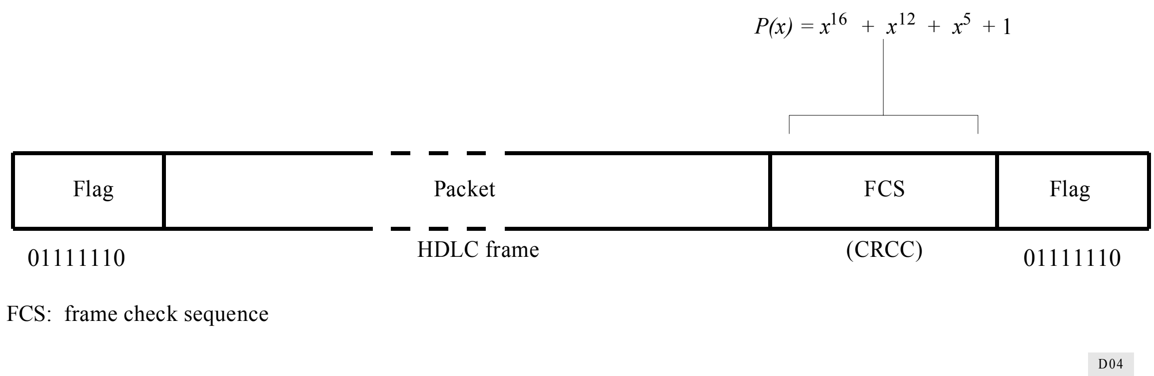
The frame level shall receive the packets and construct the frames (Fig. 1 – Step 4) which are inserted into the user data channel. The frames shall use the HDLC structure. The HDLC frame is defined in ISO/IEC 13239 (2000).

A frame (Fig. 4) comprises:

– a beginning flag: (0111 1110) (7E hex);
– a packet;
– a frame check sequence (FCS) (a cyclic redundancy check);
– an ending flag: (0111 1110) (7E hex).

When a series of packets is transmitted, the end flag of one frame may be the start flag of the next frame.

FIGURE 4
Frame level



5.2.4 Physical level

The physical level (Fig. 1 – Step 5) shall be the serial transmission of the encoded frames at a bit rate equal to the audio sampling frequency.

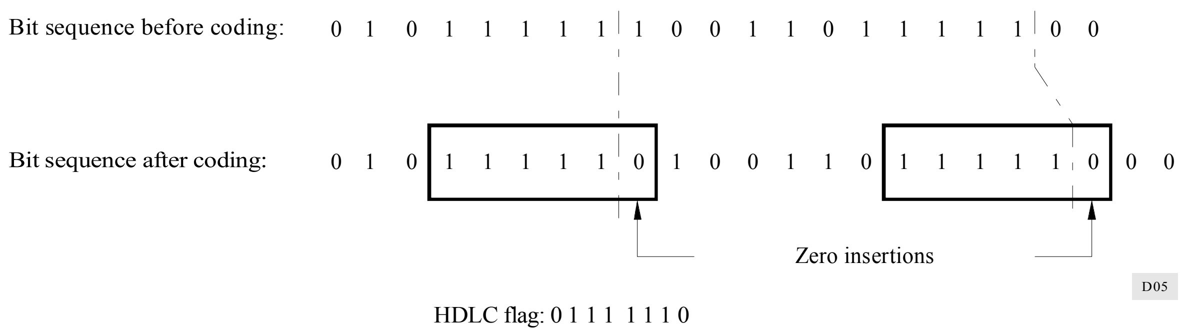
5.2.4.1 Data coding

The data in the packet part of each frame, including the FCS, shall be coded as described in ISO/IEC 13239 (2000) to insert one extra logic "0" after every sequence of five logic "1"s (Fig. 5). Sequences of six logic "1"s are therefore available to flag unambiguously the beginning and the end of a new frame, since this pattern cannot occur in data. The flag is the byte (0111 1110) (7E hex).

When there are at least seven consecutive logic "1"s the transport system shall be in the idle mode.

FIGURE 5

HDLC channel coding



5.2.4.2 Data transmission

The frames shall be transmitted in the user data channel (U) of the digital audio interface (Recommendation ITU-R BS.647) as NRZ signals.

The use of this transport system shall be signalled in the channel status channel (C) of the interface by setting channel status byte 1, bits 4-7* to code "0010".

The least significant bit of each byte shall be transmitted first.

6 Channel management

The system described so far is able to transmit messages, but no special provision is made for inserting messages in downstream equipment or for resolving any conflicts of priority. The channel management system is provided to control these functions. It defines:

– a channel formatting and block structure;

– a channel description system;

– rules for inserting the data packets.

---

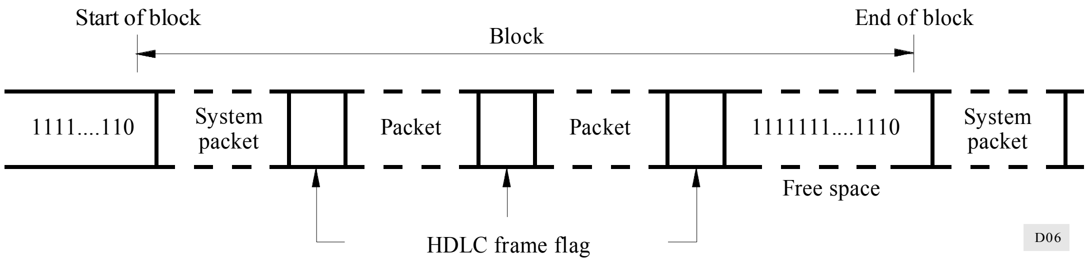
* See Recommendation ITU-R BS.647.

6.1 Channel formatting

The data flow in the user data channel shall be divided into repetitive structures called blocks. The block structure is shown diagrammatically in Fig. 6.

FIGURE 6

Block structure



6.1.1 Block rate and length

The block structure can be synchronized to an external event and the user can set the repetition rate of the blocks to suit the application. Recommended repetition rates are:

Blocks/s	Duration (ms)	Remarks
2	500	
5	200	
24	41.67	Film frame rate
25	40	PAL, SECAM or 1 250/50 HDTV frame rate
29.97	33.37	NTSC frame rate
30	33.33	1 125/60 HDTV frame rate
33.33	30	R-DAT frame rate
100	10	Shortest practical block

Any other repetition rate can, however, be defined by the user (see § 6.2.1.3). These block rates are independent of the audio sample frequency and will contain different numbers of bits at different audio sample rates.

Normally, the number of bits in a block will be constant. However, in some cases it will be impossible to provide a constant number of bits in the blocks. For instance in the case of the NTSC frame rate and an audio sampling frequency of 48 kHz, there are 8 008 audio samples every 5 video frames. Since the number 8 008 is not a multiple of 5, it is necessary to have variable block lengths.

6.1.2 Block start

A block start shall be identified by a logic "0" which follows at least seven logic "1"s and is terminated by at least seven logic "1"s which are the logic "1"s before the first logic "0" of the following block. This logic "0" is defined as the first bit of the block and the HIGH to LOW transition which follows the sequence of logic "1"s is considered as the beginning of the block (Fig. 6).

6.2 Channel description

The channel shall be divided into blocks. The start of each block is indicated by the block start sequence described in § 6.1.2.

Following the block start sequence, an optional system packet may be added which describes the block length and controls the data insertion. The system packet can have an information field which can be used to carry system data.

6.2.1 System packet

The system packet shall contain an address byte, a control byte, an optional address extension, a descriptor byte and an information field, which can contain up to 15 bytes (Fig. 7a)).

The address FF (hex) shall always be used for the system packet and shall be restricted to the system packet to permit easy identification.

6.2.1.1 Control byte of system packets

The control byte of a system packet (A_7 to A_0) (Fig. 7b) shall have the structure defined below. This is a different structure to the control bytes previously described for normal packets (§ 5.2.2.1):

bits A_7 A_6 System packet identification

1 1 This code is always used and identifies the packet as a system packet.

bit A_5 Address extension bit

0 There is no address extension.

1 There is an address extension (see § 6.2.1.2).

bit A_4 Reserved and shall not be used at present and set to logic "0".

bits A_3 A_2 A_1 A_0 Priority enable bits

A_3 Priority enable for priority 3 (highest).

A_2 Priority enable for priority 2.

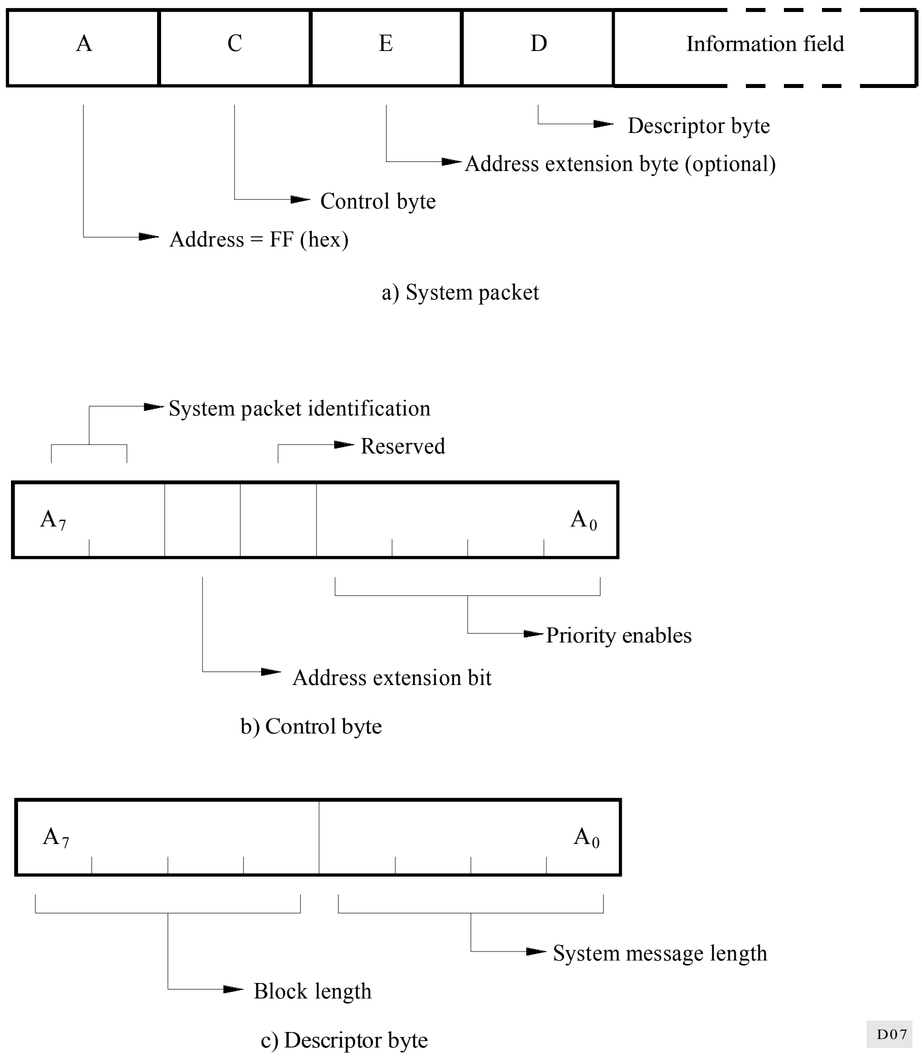
A_1 Priority enable for priority 1.

A_0 Priority enable for priority 0 (lowest).

When a priority enable bit is set to logic "1", messages which have the corresponding priority can be inserted; when the bit is set to logic "0" the insertion is forbidden. There is no restriction on the combination of enable bits which can be set. For general rules of insertion see § 6.3.1.

FIGURE 7

System packet



6.2.1.2 Address extension byte

The address extension byte shall identify the contents of the information field in the system packet and shall only be transmitted when the address extension bit is set to logic "1".

bits A_7 to A_0 Description of information field of system packet

Reserved for future use. If present, all bits should be set to logic "0".

6.2.1.3 Descriptor byte

The 8 bit descriptor byte (Fig. 7c)) shall have two fields which define the length of the block and the length of the system message.

bits $A_7\ A_6\ A_5\ A_4$ Coded length of the block

	Blocks/s	Duration (ms)	Remarks
0 0 0 0	24	41.67	Film frame rate
0 0 0 1	25	40	PAL, SECAM 1 250/50 frame rate
0 0 1 0	30	33.33	1 125/60 HDTV frame rate
0 0 1 1	29.97	33,37	NTSC frame rate
0 1 0 0	100	10	Shortest practical block
0 1 0 1	5	200	
0 1 1 0	2	500	
0 1 1 1	33.33	30	R-DAT frame rate
1 0 0 0	User defined length.		

All other codes are reserved at present and shall not be used.

bits $A_3\ A_2\ A_1\ A_0$ System message length (A_3 MSB A_0 LSB)

The number of bytes in the information field of the system packet shall be given by bits A_3-A_0.

6.2.1.4 Information field in the system packet

The information field which follows the descriptor byte can contain up to 15 bytes of data (Fig. 7). It can be used to carry time codes, to indicate the block length when it is user defined or to carry channel status data or other information. The content of the information field, if present, shall be identified by the address extension byte of the system packet (see § 6.2.1.2).

NOTE 1 – The use of the information field awaits definition of the address extension byte.

6.3 Data insertion

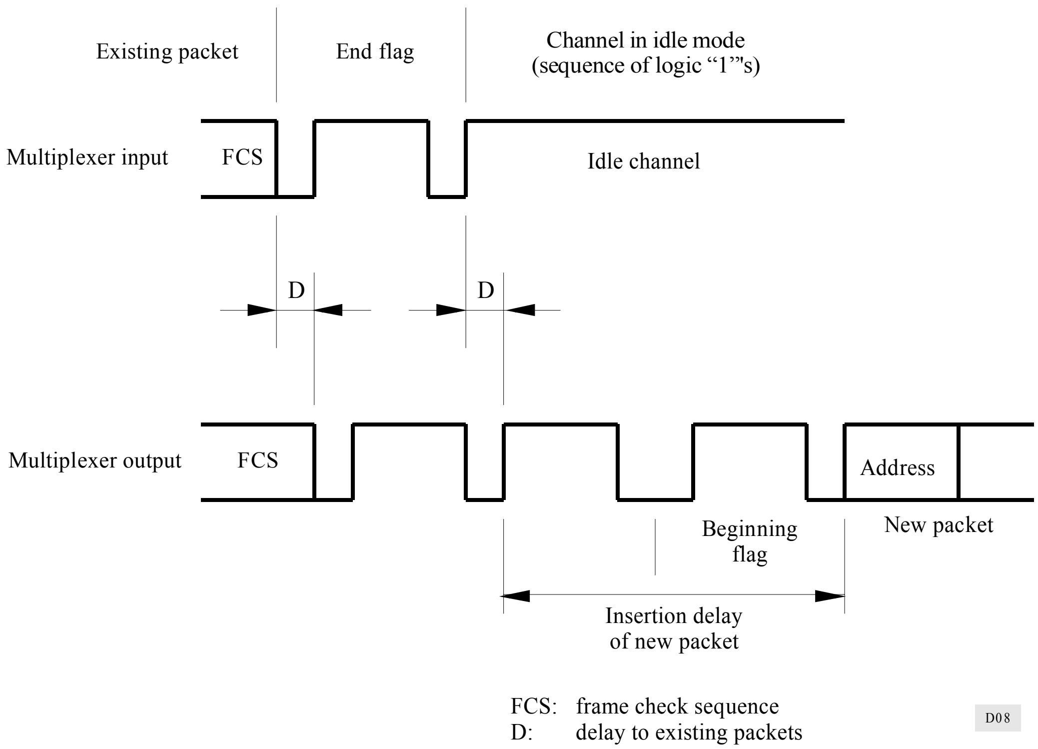
Packets shall be inserted into the blocks of the user channel by packet multiplexers. When packets are due to be inserted, a block may already contain a number of packets. Existing packets will be followed by successive logic “1”s which indicate an idle channel up to the end of the block. The ending flag of a packet and the beginning flag of the following packet can be the same, or can be separated by up to 3 bytes.

6.3.1 Rules of insertion

Multiplexing of new packets involves two steps. The first step is the detection of a block start (see § 4.1.2). The second step is the detection of seven consecutive logic “1”s which indicate the channel is in the idle mode. This allows the multiplexer to insert a new packet if there is enough free space before the end of the block. If there is space for the new packet, the seventh existing logic “1” shall be set to logic “0”, forming a new HDLC flag. The new packet (possibly including a further HDLC “beginning” flag) shall then follow (Fig. 8).

FIGURE 8

Insertion of packet



The amount of available free space depends upon the block length and the length of previously inserted messages. However, free space (justification bits) shall be reserved at the end of each block to avoid losses of packets during sampling frequency conversion or variable speed operation. The amount of free space which shall be reserved should allow the audio sample rate to be reduced to 12.5% below 48 kHz. Examples are given in Table 1 for various audio sample rates and various block lengths.

TABLE 1

Examples of numbers of justification bits for various block lengths

Sample frequency (kHz)	Free space (%)	Total bits per block			Justification bits per block		
		10 ms	40 ms	200 ms	10 ms	40 ms	200 ms
42	0	420	1 680	8 400	0	0	0
44.1	5	441	1 764	8 820	21	84	420
48	12.5	480	1 920	9 600	60	240	1 200
54	25	540	2 160	10 800	120	480	2 400

The above justification shall be mandatory over this audio sample frequency range. The interface and the transport system can be used at other audio sample frequencies, for instance, 32 kHz, but will need appropriate data management.

6.3.2 Priority management

Priority management shall be carried out at the multiplexing level and, in the first instance, the insertion of messages shall be allowed or forbidden according to the state of the priority bits of the system packet (see § 4.2.1.1). Only packets of the priorities whose priority bits are set at logic "1" shall be inserted.

The priority management rules further ensure that the multiplexers are able to average the loading of the blocks to prevent upstream multiplexers monopolizing the channel capacity.

When the application is designed, the user shall assign a priority to each message. The choice of this priority may be either:

– to ensure a maximum delay is not exceeded in the insertion of short urgent messages carrying real time data, or

– to provide a sufficient bit rate, compatible with the application, if the messages are long.

6.3.2.1 Priority insertion rules

The priority management rules shall be in accordance with Table 2. The table defines the maximum number of packets per message which can be inserted per block as a function of the block length and the priority.

TABLE 2

Maximum number of packets per message inserted per block as a function of the block length and the priority

Priority	Block length			
	10 ms	1 frame	200 ms	500 ms
	Packets per block			
3 (highest)	1	4	20	50
2	1/4	1	5	12
1	1/20	1/5	1	2
0 (lowest)	1/40	1/10	1/2	1

When the number of packets which can be inserted into a block for a given priority is greater or equal to one, packets can be multiplexed according to the rules defined in Table 2.

However, for long messages of lower priorities, where the rules will only allow one packet to be inserted into one of several blocks, extra rules are needed to prevent overloading of the earlier blocks. In these circumstances the following rules shall apply in order to spread the loading over all the available blocks.

If the insertion rules allow one packet to be introduced into every *n* blocks (*n* = 2, 4, 5, 10, 20, 40 defined in Table 2):

– the packet shall only be inserted into the first *n*/2 blocks if there is at least one of the blocks with more than half of its length still available for downstream multiplexing;

– if the packet is not inserted in the first *n*/2 blocks, it shall be inserted as soon as possible into one of the remaining *n*/2 blocks.

6.3.2.2 Real time applications

For urgent or critical messages, which are usually one packet long, the priority index can be used to control the insertion delay of the message packets. The block length represents the maximum delay needed to multiplex an urgent packet into the network (a single packet message can be inserted into a single block). This is subject to the multiplex not being overloaded at the time that the urgent or critical message originates. The block length will be chosen to give a delay which is appropriate to the application.

Table 3 shows how the maximum delay varies with the block lengths and priority.

TABLE 3

Maximum delay of packet insertion as a function of block length and priority

Priority	Block length			
	10 ms	1 frame	200 ms	500 ms
	Delay			
3 (highest)	10 ms	1 frame	200 ms	500 ms
2	1 frame	1 frame	200 ms	500 ms
1	200 ms	200 ms	200 ms	500 ms
0 (lowest)	500 ms	500 ms	500 ms	500 ms

6.3.2.3 Bit rate adaptation

The priority management rules and the choice of priority can be used to determine the number of packets that can be inserted into one block and hence the effective bit rate. For example, for a 10 ms block length and a message with priority 3 (the highest) one packet will be inserted into every block during the transmission. However with the same block length but a message with priority 2, one packet will be inserted into only one block out of every four blocks. The mean value is one packet per 40 ms.

Priority	Approximate packet/s	Approximate bits
3 (highest)	100	12 800
2	25	3 200
1	5	640
0 (lowest)	2.5	320

____