INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.223

(07/2001)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Transmission multiplexing and synchronization

Multiplexing protocol for low bit rate multimedia communication

ITU-T Recommendation H.223

(Formerly CCITT Recommendation)

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	**H.220–H.229**
Systems aspects	H.230–H.239
Communication procedures	H.240–H.259
Coding of moving video	H.260–H.279
Related systems aspects	H.280–H.299
SYSTEMS AND TERMINAL EQUIPMENT FOR AUDIOVISUAL SERVICES	H.300–H.399
SUPPLEMENTARY SERVICES FOR MULTIMEDIA	H.450–H.499

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation H.223

Multiplexing protocol for low bit rate multimedia communication

Summary

This Recommendation specifies a packet-oriented multiplexing protocol for low bit rate multimedia communication. This protocol can be used between two low bit rate multimedia terminals, or between a low bit rate multimedia terminal and a multipoint control unit or an interworking adapter. The protocol allows the transfer of any combination of digital voice/audio, digital video/image and data information over a single communication link. This protocol provides low delay and low overhead by using segmentation and reassembly and by combining information from different logical channels in a single packet. The control procedures necessary to implement this multiplexing protocol are specified in ITU-T H.245. Annexes A, B and C are multiplexing protocol extensions for low, medium and highly error-prone channels, respectively. Annex D provides an optional Reed-Solomon code error correction which is an alternative to the RCPC coding of Annex C.

Source

ITU-T Recommendation H.223 was revised by ITU-T Study Group 16 (2001-2004) and approved under the WTSA Resolution 1 procedure on 29 July 2001.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

ITU-T Recommendation H.223

Multiplexing protocol for low bit rate multimedia communication

1 General

This Recommendation specifies the frame structure, format of fields and procedures of the packet multiplexing protocol for low bit rate multimedia communication. This protocol can be used between two low bit rate multimedia terminals, or between a low bit rate multimedia terminal and a Multipoint Control Unit (MCU) or an InterWorking adapter (IWA). The control procedures necessary to implement this multiplexing protocol are specified in ITU-T H.245.

In this Recommendation communication between different protocol layers is modelled as a set of abstract primitives, which represent a logical exchange of information. The decomposition of functionality into (sub)layers, as well as the description of the primitives, do not imply a particular method of implementation. In particular, layers may exchange the contents of a logical unit (an SDU) in a "streaming" mode where information exchange may start before the transferring layer has the complete unit in its possession.

2 Normative references

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

[1] ITU-T H.245 (2000), *Control protocol for multimedia communication*.

[2] ITU-T V.42 (1996), *Error-correcting procedures for DCEs using asynchronous-to-synchronous conversion*.

[3] ITU-T H.324 (1998), *Terminal for low bit rate multimedia communication*.

[4] ITU-T Q.922 (1992), *ISDN data link layer specification for frame mode bearer services*.

3 Definitions and format conventions

3.1 Definition of terms

This Recommendation defines the following terms:

3.1.1 adaptation layer (AL): The upper of the two layers of the multiplexer of this Recommendation.

3.1.2 AL-PDU: An information unit exchanged between peer Adaptation Layer entities. An AL-PDU is conveyed as one MUX-SDU.

3.1.3 AL-SDU: A logical information unit whose integrity is preserved in transfer from one AL user to the peer AL user.

3.1.4 AL user: A higher-layer entity which makes use of the services of the Adaptation Layer.

3.1.5 control channel: A logical channel which carries H.245 control messages.

3.1.6 header error control (HEC) field: A 3-bit CRC field in the MUX-PDU header which is used to detect errors that affect the MC field.

3.1.7 logical channel number (LCN): A unique integer between 0 and 65535 assigned to a logical channel.

3.1.8 multiplex code (MC) field: A 4-bit field in the MUX-PDU header which specifies, by reference to a multiplex table entry, the logical channel to which each octet in the information field belongs.

3.1.9 multiplex (MUX) layer: The lower of the two layers of the multiplexer of this Recommendation.

3.1.10 multiplex table: A table with up to 16 entries which specifies the multiplexing pattern for the information field of a MUX-PDU.

3.1.11 MUX-PDU: An information unit exchanged between peer MUX layer entities.

3.1.12 MUX-SDU: A logical information unit whose integrity is preserved in transfer from one Adaptation Layer to the peer Adaptation Layer.

3.1.13 non-segmentable logical channel: A logical channel whose MUX-SDUs may not be segmented. MUX-SDUs from a non-segmentable logical channel are transmitted in consecutive octets of a single MUX-PDU.

3.1.14 packet marker (PM) field: A one-bit field used to mark the end of a MUX-SDU from a segmentable logical channel.

3.1.15 protocol data unit (PDU): A unit of information exchanged between peer protocol layer entities.

3.1.16 quality of service (QOS): The quality of the service that individual information streams receive from the multiplexer, as measured by parameters such as bit rate, delay jitter, loss, etc.

3.1.17 segmentable logical channel: A logical channel whose MUX-SDUs may be segmented. Segmentation allows the temporary suspension of the transmission of a MUX-SDU in order to transmit octets from another MUX-SDU.

3.1.18 service data unit (SDU): A logical unit of information whose integrity is preserved in transfer from one protocol layer entity to the peer protocol layer entity.

3.1.19 slot: A consecutive sequence of octets within a single MUX-PDU, described by a single H.245 MultiplexElement structure of type logicalChannelNumber. Each slot holds an integral number of octets from a single MUX-SDU.

3.2 Format conventions

The numbering, field mapping and bit transmission conventions used in this Recommendation are consistent with those used in ITU-T V.42.

3.2.1 Numbering convention

The basic numbering convention used in this Recommendation is illustrated in Figure 1. The bits in each information unit are grouped into octets. The bits of an octet are shown horizontally and are numbered from 1 to 8. Multiple octets are shown vertically and are numbered from 1 to n.

3.2.2 Order of bit transmission

The octets are transmitted in ascending numerical order; inside an octet, bit 1 is the first bit to be transmitted.

3.2.3 Field mapping convention

When a field is contained within a single octet, the lowest-numbered bit of the field represents the lowest-order value (or the least significant bit).

When a field spans more than one octet, the highest-numbered bit of the first octet represents the highest-order value, and the lowest-numbered bit of the last octet represents the lowest-order value.

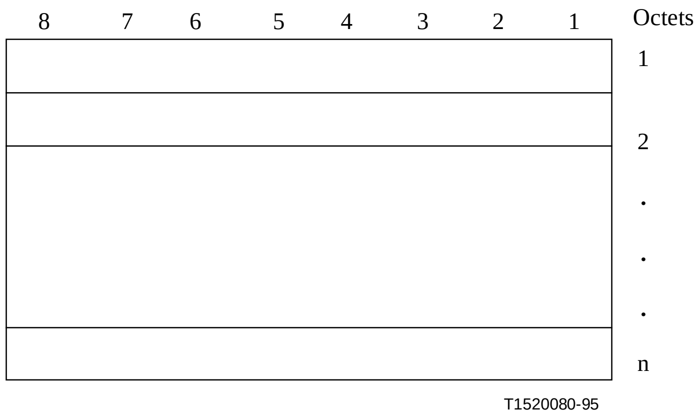


Figure 1/H.223 – Format convention

An exception to the preceding field-mapping convention is the Cyclic Redundancy Check (CRC) field. In this case, the lowest-numbered bit of the first octet is the highest-order term of the polynomial representing the CRC field; the highest-numbered bit of the last octet is the lowest-order term of the polynomial representing the CRC field.

4 Abbreviations

This Recommendation uses the following abbreviations:

AL	Adaptation Layer
AL1-AL3	Adaptation Layer 1-3
CRC	Cyclic Redundancy Check
DRTX	Declined Retransmission
EI	Error Indication
HDLC	High-level Data Link Control
HEC	Header Error Control
IWA	InterWorking Adapter
LAPM	Link Access Procedure for Modems
LCN	Logical Channel Number
MC	Multiplex Code
MUX	Multiplex
PDU	Protocol Data Unit
PM	Packet Marker
PT	Payload Type

QoS Quality of Service

SDU Service Data Unit

SN Sequence Number

SREJ Selective Reject

5 Overview

This Recommendation specifies a packet-oriented multiplexing protocol designed for the exchange of one or more information streams between higher-layer entities such as data and control protocols and audio and video codecs.

In this Recommendation, each information stream is represented by a uni-directional logical channel which is identified by a unique Logical Channel Number (LCN), an integer between 0 and 65535. LCN0 is a permanent logical channel assigned to the H.245 control channel. All other logical channels are dynamically opened and closed by the transmitter using the H.245 OpenLogicalChannel and CloseLogicalChannel messages. All necessary attributes of the logical channel are specified in the OpenLogicalChannel message. For applications that require a reverse channel, a procedure for opening bi-directional logical channels is also defined in ITU-T H.245.

The general structure of the multiplexer is shown in Figure 2. The multiplexer consists of two distinct layers: a Multiplex (MUX) layer and an Adaptation Layer (AL).

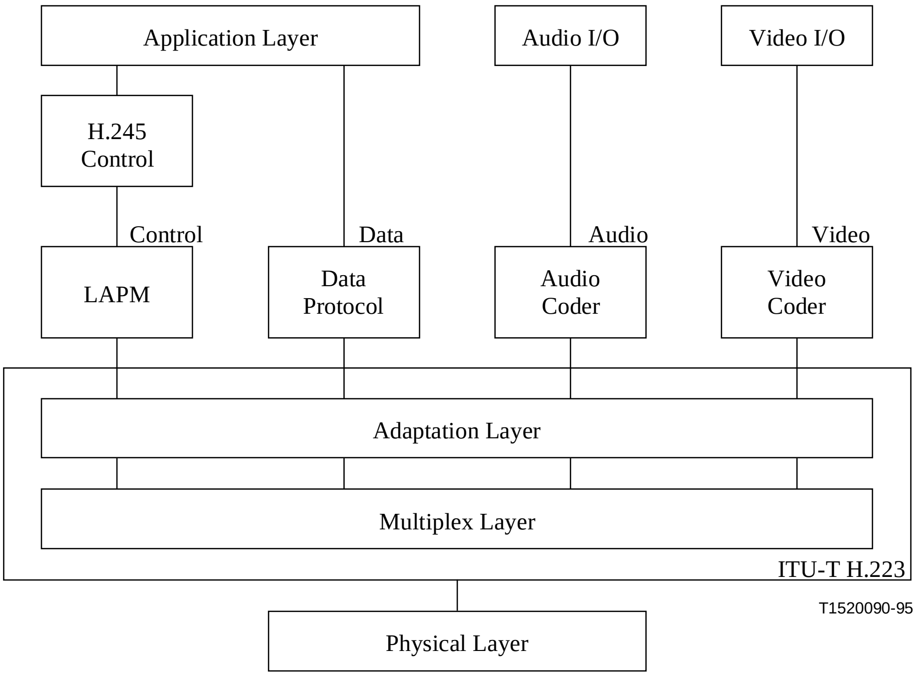


Figure 2/H.223 – Protocol stack for ITU-T H.223

5.1 Multiplex (MUX) layer overview

The MUX layer is responsible for transferring information received from the AL to the far end using the services of an underlying physical layer. The MUX layer exchanges information with the AL in logical units called MUX-SDUs. MUX-SDUs always contain an integral number of octets that belong to a single logical channel. MUX-SDUs typically represent information blocks whose start and end marks the location of fields which need to be interpreted in the receiver.

MUX-SDUs are transferred by the MUX layer to the far end in one or more variable-length packets called MUX-PDUs. MUX-PDUs consist of a one-octet header, followed by a variable number of octets in the information field. MUX-PDUs are delimited by HDLC flags. The HDLC zero-bit insertion method is used to ensure that a flag is not simulated within the MUX-PDU.

Octets from multiple logical channels may be present in a single MUX-PDU information field. The header octet contains a 4-bit Multiplex Code (MC) field which specifies, by reference to a multiplex table entry, the logical channel to which each octet in the information field belongs. Multiplex table entry 0 is permanently assigned to the control channel. Other multiplex table entries are formed by the transmitter and are signalled to the far end via the control channel prior to their use.

Multiplex table entries specify a pattern of slots each assigned to a single logical channel. Any one of 16 multiplex table entries may be used in any given MUX-PDU. This allows rapid, low-overhead switching of the number of bits allocated to each logical channel from one MUX-PDU to the next. The construction of multiplex table entries and their use in MUX-PDUs is entirely under the control of the transmitter, subject to certain receiver capabilities.

When a logical channel is opened, it is designated to be either non-segmentable or segmentable. MUX-SDUs from segmentable logical channels may be broken into segments which are then transferred to the far end in one or more MUX-PDUs. Such segmentation is useful in providing improved Quality of Service (QoS), for example, by allowing the temporary suspension of the transmission of a long MUX-SDU from a segmentable data logical channel, in order to transmit a MUX-SDU from a non-segmentable audio logical channel.

5.2 Adaptation layer overview

The unit of information exchanged between the AL and the higher-layer AL users is an AL-SDU. The method of mapping information streams from higher layers into AL-SDUs is outside the scope of this Recommendation, and is specified in the System Recommendation that uses H.223.

AL-SDUs contain an integer number of octets. The AL adapts AL-SDUs to the MUX layer by adding, where appropriate, additional octets for purposes such as error detection, sequence numbering and retransmission. The information unit exchanged between peer AL entities is called an AL-PDU. An AL-PDU is conveyed as one MUX-SDU.

Three different types of ALs, named AL1 through AL3, are specified in this Recommendation:

- AL1 is designed primarily for the transfer of data or control information. AL1 does not provide any error control; all necessary error protection should be provided by the AL1 user.

  In the framed transfer mode, AL1 receives information from its higher layer (e.g. a data link layer protocol such as LAPM/V.42 or LAPF/Q.922, which provides error control) in variable-length AL-SDUs, and simply passes these to the MUX layer in MUX-SDUs without any modifications.

  In the unframed mode, AL1 is used to transfer an unframed sequence of octets from an AL1 user. In this mode, one AL-SDU represents the entire sequence and is assumed to continue indefinitely.

- AL2 is designed primarily for the transfer of digital audio.

  AL2 receives information from its higher layer (e.g. an audio encoder) in AL-SDUs, possibly of variable-length and passes these to the MUX layer in MUX-SDUs, after adding 1 octet for an 8-bit CRC, and optionally adding 1 octet for sequence numbering.

- AL3 is designed primarily for the transfer of digital video.

  AL3 receives information from its higher layer (e.g. a video encoder) in variable-length AL-SDUs and passes these to the MUX layer in MUX-SDUs, after adding 2 octets for a 16-bit CRC, and optionally adding 1 or 2 control octets. AL3 includes a retransmission protocol designed for video.

6 Multiplex (MUX) layer specification

6.1 Framework of MUX layer

The MUX layer provides the capabilities to transfer MUX-SDUs from the sending AL to the receiving AL using the services of a physical layer below. MUX-SDUs shall always contain an integral number of octets. MUX-SDUs that belong to a given logical channel shall be transferred by the MUX layer in the same order they are received from the AL above.

6.2 Primitives exchanged between the MUX layer and the AL

The MUX layer may interface with one or more ALs. The information exchanged between the MUX layer and each individual AL includes the following primitives:

- MUX-DATA.request (MUX-SDU).
- MUX-DATA.indication (MUX-SDU).
- MUX-Abort.request.
- MUX-Abort.indication

6.2.1 Description of primitives

- MUX-DATA.request: This primitive is issued to the MUX layer by an AL sending entity to request the transfer of a MUX-SDU to the corresponding receiving entity.
- MUX-DATA.indication: This primitive is issued by the MUX layer to an AL receiving entity to indicate the arrival of a MUX-SDU from the corresponding sending entity.
- MUX-Abort.request: This primitive is issued to the MUX layer by an AL sending entity to signal that a partially delivered MUX-SDU is to be discarded. This primitive may be used by all AL types.
- MUX-Abort.indication: This primitive is issued by the MUX layer to an AL receiving entity to signal that a partially delivered MUX-SDU is to be discarded.

6.2.2 Description of parameter

- MUX-SDU: This parameter contains the information of an integral number of octets from or to an AL. A MUX-SDU shall contain exactly one complete AL-PDU.

6.3 MUX-PDU framing

All MUX-PDUs shall be delimited using HDLC flags.

6.3.1 Flag

All MUX-PDUs shall be preceded and followed by the flag consisting of the unique bit pattern "01111110". The flag preceding the MUX-PDU is defined as the opening flag. The flag following the MUX-PDU is defined as the closing flag. The closing flag may also serve as the opening flag of the next MUX-PDU. However, all receivers conforming to this Recommendation shall accommodate receipt of more than one consecutive flag, as the flag may be transmitted repetitively between MUX-PDUs.

6.3.1.1 Transparency

The transmitter shall examine the MUX-PDU content between the opening and closing flags, and shall insert a "0" bit after all sequences of five contiguous "1" bits to ensure that a flag is not simulated within the MUX-PDU. The receiver shall examine the received bitstream between the opening and closing flags and shall discard any "0" bit which directly follows five contiguous "1" bits.

6.4 MUX-PDU format and coding

All MUX-PDUs shall conform to the format shown in Figure 3.

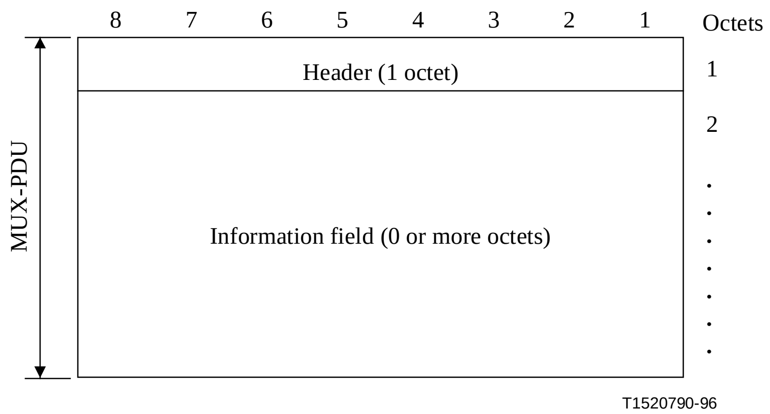
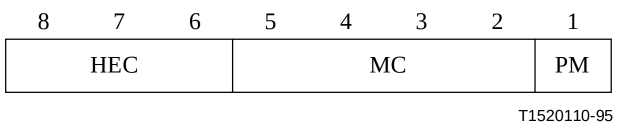


Figure 3/H.223 – MUX-PDU format

6.4.1 Header

The format of the header shall conform to the format shown in Figure 4.



MC Multiplex Code
HEC Header Error Control
PM Packet Marker

Figure 4/H.223 – Header format of the MUX-PDU

6.4.1.1 Multiplex Code (MC) field

The 4-bit MC field specifies to which logical channel each octet of the MUX-PDU information field belongs, by referencing an entry in the multiplex table. The field represents the number of the multiplex table entry, from 0 to 15. Multiplex table entry 0 is permanently assigned to the control channel, and shall always represent a pattern of octets assigned to the control channel (LCN0) which continue until the closing flag. Prior to their use, multiplex table entries 1 to 15 are sent to the far end in H.245 MultiplexEntrySend messages, according to the procedure and syntax described in ITU-T H.245.

Unless specified otherwise in the System Recommendation, at the start of communication, only table entry 0 is available, and table entries 1 to 15 are deactivated. The multiplex table entries used in each direction of transmission are independent of each other, and may be different.

Receivers should discard any MUX-PDU whose MC field references a deactivated multiplex table entry. Receivers should also discard any MUX-PDU which contains octets for a logical channel which is not open.

All receivers conforming to this Recommendation shall signal their capability to receive and correctly interpret either basic or enhanced MultiplexEntryDescriptors (for definitions, see ITU-T H.245) using the receive capability indication h223MultiplexTableCapability, specified in ITU-T H.245.

Receivers signalling basic h223MuxTableCapability shall be capable of receiving and correctly interpreting MultiplexEntryDescriptors which satisfy the following constraints:

- Maximum elementList size: 2
- Maximum nesting depth: 1
- Maximum subElementList size: 2

and whose first MultiplexElement in the elementList does not use a non-segmentable logical channel more than once and whose second MulitplexElement in the elementList uses only segmentable logical channels.

Receivers signalling enhanced capability shall signal their capability to receive and correctly interpret MultiplexEntry Descriptors, according to the H.245 h223MultiplexTableCapability indication. A receiver signalling enhanced capability is also capable of receiving and correctly interpreting all MultiplexEntryDescriptors that are included in the basic capability.

NOTE – The MC field of each MUX-PDU should be selected in a manner that provides each information stream the Quality of Service (QoS) it needs. This is the responsibility of a local multiplexing implementation which is outside the scope of this Recommendation.

6.4.1.2 Header Error Control (HEC) field

The 3-bit HEC field provides error detection capabilities over the MC field using a 3-bit CRC.

The HEC field shall contain the remainder of the division (modulo 2) by the generator polynomial $P(x) = x^3 + x + 1$ of the product x^3 multiplied by the content of the MC field. The polynomial representing the content of the MC field is generated using bit number 2 (i.e. the least significant bit) in the MC field as the coefficient of the highest-order term. The polynomial representing the content of the CRC field is generated using bit number 6 (i.e. the least significant bit) as the coefficient of the highest-order term. Table 1 shows the values of the 3-bit HEC field as a function of the 4-bit MC field.

Receivers should discard any MUX-PDU whose HEC field fails the error check.

6.4.1.3 Packet Marker (PM) field

The 1-bit PM field shall be used to mark the end of MUX-SDUs of segmentable logical channels, as described in 6.5.

Table 1/H.223 – Values of the HEC field as a function of the values of the MC field

MC Field	HEC Field
Bit number 5 4 3 2	**Bit number 8 7 6**
0 0 0 0	0 0 0
0 0 0 1	1 0 1
0 0 1 0	1 1 1
0 0 1 1	0 1 0
0 1 0 0	0 1 1

Table 1/H.223 – Values of the HEC field as a function of the values of the MC field

MC Field	HEC Field
Bit number **5 4 3 2**	**Bit number** **8 7 6**
0 1 0 1	1 1 0
0 1 1 0	1 0 0
0 1 1 1	0 0 1
1 0 0 0	1 1 0
1 0 0 1	0 1 1
1 0 1 0	0 0 1
1 0 1 1	1 0 0
1 1 0 0	1 0 1
1 1 0 1	0 0 0
1 1 1 0	0 1 0
1 1 1 1	1 1 1

6.4.2 Information field

The multiplex table entry selected by the MC field specifies the multiplexing pattern for the information field, according to the multiplex table entry syntax described in ITU-T H.245. Octets from multiple logical channels may be present in the information field. The information field may be terminated on any octet boundary by closing the MUX-PDU with a closing flag, except a MUX-SDU from a non-segmentable logical channel shall not be interrupted.

The procedure given in this paragraph is optional and is used only when it is required by the System Recommendation that uses ITU-T H.223: when this option is used, the transmitter shall exclusive-OR each octet in the information field with the octet 000uxyz0 prior to applying the transparency procedure where "uxyz" represent the bits of the MC field where z corresponds to the least significant bit (bit number 2) of the field. The receiver shall perform the same operation to restore the original information field content. This procedure is followed to ensure that errors affecting the MC field will alter the received information field octets, with high probability ensuring the failure of any CRC checks applied to the information field content.

NOTE 1 – When the above procedure is not used, the multiplexer should be designed such that in case of undetected errors affecting the MC field, the failure of any CRC checks applied to the information field is ensured with high probability.

The length of the information field is not limited; however, transmitters should consider the error characteristics of the underlying physical medium when choosing the length of the information field. In the event of bit errors that affect the MC field, the entire MUX-PDU may be lost.

NOTE 2 – In the receiver, the MUX layer may pass the octets of the information field to the AL in a "streaming" mode before having the complete MUX-PDU in its possession.

6.4.3 Abort

A MUX-PDU which has no information field shall be interpreted in the receiver as an abort, if its PM field is set to "0" and its MC field is the same as that of the previously received MUX-PDU. The MUX-SDU to be aborted is the one which occupied the last octet in the previously received MUX-PDU.

6.5 Marking of MUX-SDU boundaries

It is necessary to detect the boundaries of MUX-SDUs in the receiver in order to identify the location of all fields which the receiver must interpret in the AL and/or in a frame-oriented higher-layer. This shall be accomplished as follows:

For non-segmentable logical channels, each MUX-SDU shall begin coincident with a slot specified in a single MultiplexElement structure whose type is logicalChannelNumber (see ITU-T H.245), and shall end after the specified repeatCount, or at the closing flag of the current MUX-PDU, whichever occurs first. The actual length of the MUX-SDU may be smaller than the length of the slot, provided that the current MUX-PDU is terminated by a closing flag immediately after the MUX-SDU. Since the size of each MUX-SDU may vary, multiple multiplex table entries may be defined to match the possible lengths of MUX-SDUs, in order to mix these MUX-SDUs with octets from other logical channels. It should be noted that the definitions given here together with the conditions given in ITU-T H.245 imply that it is allowed to place more than one MUX-SDU from a non-segmentable logical channel in one MUX-PDU, but only when the remote receiver has indicated the enhanced multiplex capability.

For segmentable logical channels, each MUX-SDU may be broken into segments and these segments may be transferred in one or more MUX-PDUs. The PM field in the MUX-PDU header shall be used to mark the end of each MUX-SDU. Specifically, the PM field shall be set to "1" to indicate that the last octet of the previous MUX-PDU was the final octet of the terminating MUX-SDU. As a result of this procedure, only one segmentable MUX-SDU is permitted to terminate within a MUX-PDU; as soon as the end of any MUX-SDU from a segmentable logical channel is reached, the MUX-PDU shall be terminated with a closing flag and the PM field in the next MUX-PDU shall be set to "1". In all other circumstances, the PM field shall be set to "0". Another result of this procedure is that a MUX-PDU will never contain octets from two different MUX-SDUs of the same segmentable logical channel.

An empty MUX-PDU with no information field shall be transmitted for the purpose of terminating a MUX-SDU from a segmentable logical channel, if the transmitter has no information to send immediately after closing the MUX-PDU. The PM field of this MUX-PDU shall be set to "1", and the MC field shall be the same as that of the previous MUX-PDU.

6.6 Examples

Table 2 includes examples of MultiplexEntryDescriptors which include 1, 2 or 3 MultiplexElements in the elementList. Each row in the table corresponds to a MultiplexEntryDescriptor. For each MultiplexEntryDescriptor, the number of MultiplexElements in the elementList, the nesting depth and the subelementList size are given in separate columns.

Five logical channels are assumed as follows: LCN0: control, LCN1: audio I, LCN2: data, LCN3: video, LCN4: audio II. Audio logical channels are designated as non-segmentable, and all others are designated as segmentable.

The first five rows show examples of basic MultiplexEntryDescriptors:

The first two rows show how the entire MUX-PDU information field can be assigned to a single logical channel. It should be noted that the entry shown in row 1 can be used to send audio MUX-SDUs of any length, but not to send more than one audio MUX-SDU.

The third row illustrates how video may be transmitted after an audio MUX-SDU in a single MUX-PDU.

The fourth row shows how data and video may be mixed using a repeating pattern of 1 octet of data and 3 octets of video.

The fifth row shows how a short audio MUX-SDU, possibly representing background noise information sent during a silence period, may be mixed using a repeating pattern of data and video. This entry is used later in this clause to illustrate the construction of the information field.

The last three rows show examples of enhanced MultiplexEntryDescriptors:

The sixth row illustrates how audio may be mixed with octets from video, data and control channels.

The seventh row shows a MultiplexEntryDescriptor with 3 MultiplexElements used to send two audio MUX-SDUs from two different audio logical channels, mixed with octets from data and video channels.

Finally, the eighth row shows an example of 2-level nesting, where an audio MUX-SDU is followed by an alternating pattern of data and video octets which repeats five times, and where the entire pattern including the audio MUX-SDU repeats untilClosingFlag.

Table 2/H.223 – Examples of MultiplexEntryDescriptors (LCN: logicalChannelNumber, RC: repeat Count, UCF: untilClosingFlag)

Row	MultiplexEntryDescriptor	Element ListSize	Nesting Depth	Subelement ListSize	Example
1	{LCN1,RC UCF}	1	0	0	All audio
2	{LCN3,RC UCF}	1	0	0	All video
3	{LCN1,RC21},{LCN3,RC UCF}	2	0	0	Audio, All video
4	{{LCN2,RC1},{LCN3,RC3}, RC UCF}	1	1	2	1:3 data video
5	{LCN1,RC4},{{LCN2,RC1}, {LCN3,RC2},RC UCF}	2	1	2	Audio, 1:2 data video
6	{LCN1,RC21},{{LCN2,RC2}, {LCN3,RC6},{LCN0,RC1}RC UCF}	2	1	3	Audio, 2:6:1 data video control
7	{LCN1,RC21},{LCN4,RC25}, {{LCN2,RC1},{LCN3,RC1} RC UCF}	3	1	2	Audio I, Audio II, 1:1 data video
8	{{LCN1,RC25},{{LCN2,RC1}, {LCN3,RC1},RC5},RC UCF}	1	2	2	2-level nesting

Figure 5 shows an example of the construction of the information field from the MultiplexEntryDescriptor and illustrates the use of the PM field. The MultiplexEntryDescriptor shown in row 5 of Table 2 is used in this example.

Suppose that in this example at one instant of time, the multiplexer has three MUX-SDUs ready for transmission: a 4-octet MUX-SDU from LCN1, a 3-octet MUX-SDU from LCN2 and a 3-octet MUX-SDU from LCN3.

The MUX-PDU is formed starting with the 4-octet MUX-SDU from LCN1, and continuing with a 1-octet segment from LCN2, a 2-octet segment from LCN3, a 1-octet segment from LCN2 and another 1-octet segment from LCN3. Since the end of the MUX-SDU from LCN3 is then reached, the MUX-PDU is closed with a flag, and the PM field in the next MUX-PDU is set. The last octet of the MUX-SDU from LCN2 may be transmitted in any of the following MUX-PDUs.

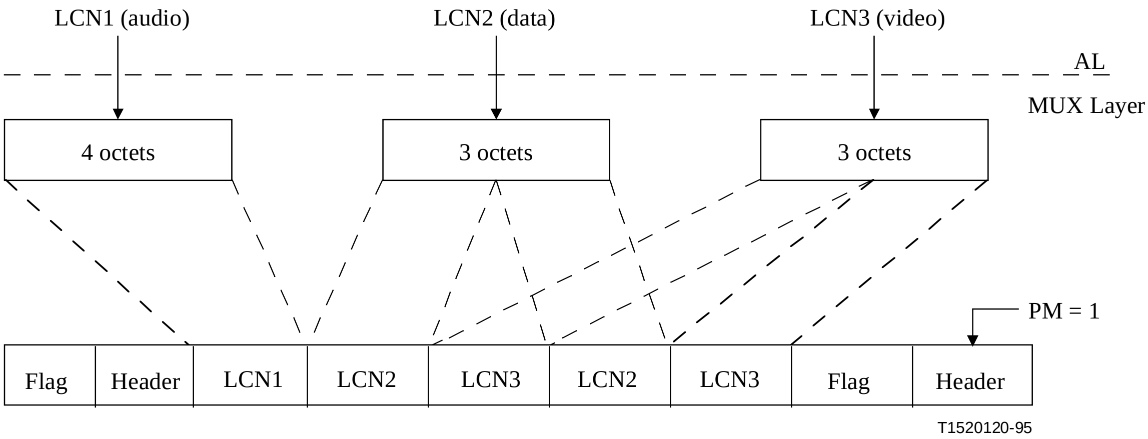


Figure 5/H.223 – Information field example

7 Adaptation Layer (AL) specification

7.1 Introduction

This clause describes the interactions between the AL and the higher layer above, and the AL and the MUX layer, as well as end-to-end operations between peer ALs. The Adaptation Layer (AL) enhances the services provided by the underlying MUX layer to support functions required by AL users and to support the mapping between the MUX layer and the layer above. Three different types of ALs, named AL1, AL2 and AL3, are specified.

The AL is selected by the transmitter using the OpenLogicalChannel message of ITU-T H.245 at the time a logical channel is opened. Any one of the three types of ALs may be used to carry a given logical channel subject to restrictions that may be imposed by the System Recommendation that uses ITU-T H.223. The AL includes a few optional fields which are selected by the transmitter in the OpenLogicalChannel message at the time a logical channel is opened.

The unit of information exchanged between the AL and a higher-layer entity is called an AL-SDU. AL-SDUs may be of variable length. The maximum length of AL-SDUs is determined by the AL user. The method of mapping the information stream from the higher layer to AL-SDUs is outside the scope of this Recommendation, and is defined in the System Recommendation (e.g. ITU-T H.324) that uses ITU-T H.223. AL-SDUs that belong to a given logical channel shall be transferred by the AL in the same order they are received from the higher-layer entity. AL transfers a complete AL-SDU received from the AL user in a single AL-PDU. An AL-PDU is mapped directly to a single MUX-SDU, the parameter of the MUX layer primitive, and vice versa.

7.2 Adaptation Layer Type 1 (AL1) specification

7.2.1 Framework of AL1

AL1 is designed primarily for the transfer of data or control information.

AL1 does not provide any error detection or correction capability. Therefore, any necessary error control, possibly including a retransmission procedure, should be provided by the higher layer.

AL1 provides two transfer modes:

a) framed transfer mode; and

b) unframed transfer mode.

In the framed transfer mode, AL1 may be used to transfer frames generated by a higher-layer protocol such as the data link layer protocol LAPM/V.42 or LAPF/Q.922. In this case, frames are first mapped to AL-SDUs and these are then passed by AL1 in MUX-SDUs to the MUX layer.

AL1 may also be used to carry an unframed octet sequence. In this mode, any internal framing present in the octet sequence is not visible to AL1 which passes the octets received from the higher layer to the MUX layer without paying any attention to framing.

The transfer mode of AL1 is selected by the transmitter in the H.245 OpenLogicalChannel message.

Logical channels which are transferred by AL1 using the unframed transfer mode shall be designated as segmentable, so that the octet transmission can be interrupted to send octets from other information streams. However, since the AL-SDU continues indefinitely, for such logical channels, the PM field shall never be set to "1".

7.2.2 Primitives exchanged between AL1 and AL1 user

The information exchanged between AL1 and the AL1 user includes the following primitives:

- AL-DATA.request (AL-SDU).
- AL-DATA.indication (AL-SDU).
- AL-Abort.request.
- AL-Abort.indication.

7.2.2.1 Description of primitives

- AL-DATA.request: This primitive is issued by an AL1 user to AL1 to request the transfer of an AL-SDU to its corresponding receiving entity.
- AL-DATA.indication: This primitive is issued to an AL1 user by AL1 to indicate the arrival of an AL-SDU.
- AL-Abort.request: This primitive is issued to AL1 by an AL1 user to signal that a partially delivered AL-SDU is to be aborted. This primitive is not used in the unframed transfer mode.
- AL-Abort.indication: This primitive is issued by AL1 to an AL1 user to signal that a partially delivered AL-SDU is to be aborted. This primitive is not used in the unframed transfer mode.

7.2.2.2 Description of parameter

- AL-SDU: This parameter specifies the unit of information exchanged between AL1 and the AL1 user. Each AL-SDU shall contain an integral number of octets. The length of AL-SDUs may be variable. The maximum size of AL-SDUs shall be determined by the AL1 user. The octets in an AL-SDU are numbered from 1 to n, and in each octet, bits are numbered from 1 to 8. Bit 1 of octet 1 is transmitted first.

7.2.3 Procedures for Abort

Abort procedures may be used when information is exchanged between layers in a streaming mode.

When an AL-Abort.request primitive is sent from the AL1 user to AL1 in order to abort a partially delivered AL-SDU, AL1 shall immediately send a MUX-Abort.request primitive to the MUX layer, if a MUX-SDU containing that AL-SDU has already been partially delivered to the MUX layer.

In the AL1 receiver, when a MUX-Abort.indication primitive is received from the MUX layer, AL1 shall immediately send an AL-Abort.indication primitive to the AL1 user, if that AL-SDU has already been partially delivered to the AL1 user.

Abort procedures shall not be used in the unframed transfer mode.

7.3 Adaptation Layer Type 2 (AL2) specification

7.3.1 Framework of AL2

AL2 is designed primarily for the transfer of digital audio.

AL2 provides an 8-bit CRC for error-detection. AL2 also supports optional sequence numbering which may be used to detect missing and misdelivered AL-PDUs. AL2 transfers variable-length AL-SDUs of integral number of octets.

7.3.2 Primitives exchanged between AL2 and AL2 user

The information exchanged between AL2 and an AL2 user includes the following primitives:

- AL-DATA.request (AL-SDU).
- AL-DATA.indication (AL-SDU, EI).
- AL-Abort.request.

7.3.2.1 Description of primitives

- AL-DATA.request: This primitive is issued by an AL2 user to AL2 to request the transfer of an AL-SDU to a corresponding AL2 user.
- AL-DATA.indication: This primitive is issued to an AL2 user by AL2 to indicate the arrival of an AL-SDU.
- AL-Abort.request: This primitive is issued to AL2 by an AL2 user to signal that a partially delivered AL-SDU is to be aborted.

7.3.2.2 Description of parameters

- AL-SDU: This parameter specifies the unit of information exchanged between AL2 and the AL2 user. Each AL-SDU shall contain an integral number of octets. The length of AL-SDUs may be variable. The maximum length of AL-SDUs that an AL2 receiver can accept shall be signalled via the H.245 control channel. The octets in an AL-SDU are numbered from 1 to n, and in each octet, bits are numbered from 1 to 8. Bit 1 of the octet 1 is transmitted first. An AL2 receiving entity may deliver an empty AL-SDU to the AL2 user to indicate that an AL-SDU is missing.
- Error Indication (EI): This parameter may be used in the AL2 receiver to pass error indications to the AL2 user. The precise procedures for using this parameter, and its numerical coding, are outside the scope of this Recommendation.

7.3.3 Functions, format and coding of AL2

7.3.3.1 Functions of AL2

AL2 provides the following functions:

- Error detection and indication;
- Optional sequence numbering.

7.3.3.2 Format and coding of AL2

The format of the AL-PDU is illustrated in Figure 6.

7.3.3.2.1 Sequence Number (SN) field

The optional 8-bit SN provides a capability for sequencing AL-PDUs. The sequence number may be used by the AL2 receiving entity to detect missing and misdelivered AL-PDUs.

All receivers conforming to this Recommendation shall be capable of receiving and correctly interpreting AL-PDUs that include the SN field. The use of the SN field shall be determined by the transmitter and shall be signalled to the far end in the OpenLogicalChannel message of ITU-T H.245.

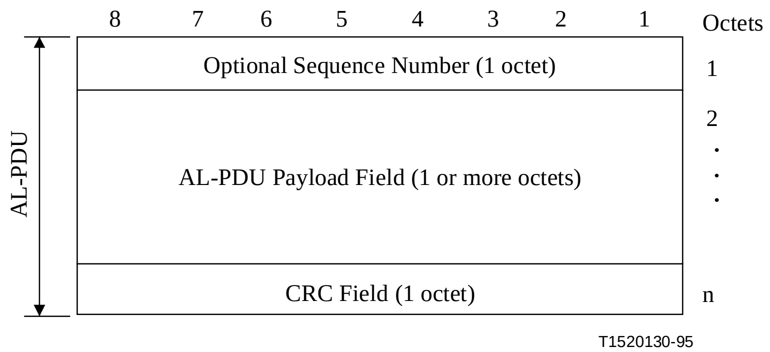


Figure 6/H.223 – AL-PDU format for AL2

When the SN field is in use, the AL2 receiver may detect that an AL-PDU is missing or has been misdelivered by the MUX layer. The AL2 receiver should discard any misdelivered AL-PDUs that it detects.

7.3.3.2.2 AL-PDU payload field

The AL-PDU payload field contains a complete AL-SDU, where the first octet corresponds to the first octet of the AL-SDU.

7.3.3.2.3 CRC field

The 8-bit CRC provides an error detection capability across the entire AL-PDU.

The 8-bit CRC field shall contain the remainder of the division (modulo 2) by the generator polynomial $p(x) = x^8 + x^2 + x + 1$ of the product x^8 multiplied by the content of the AL-PDU, excluding the CRC field, and including the SN field, if it is used. The polynomial representing the content of the AL-PDU is generated using bit number 1 of the first octet as the coefficient of the highest-order term.

As a typical implementation in the transmitter, the initial content of the register of the device computing the remainder of the division is preset to all 0's and is then modified by division by the generator polynomial (as described above) of the content of the AL-PDU, not including the bits in the CRC field; the resulting remainder is transmitted as the 8-bit CRC. The coefficient of the highest-order term of the remainder polynomial corresponds to bit number 1 of the CRC field.

NOTE – In contrast to the CRC procedure used for the 16-bit CRC in AL3, the CRC procedure used here does not include any pre- or post-conditioning.

7.3.4 Procedures for Abort

Abort procedure may be used when information is exchanged between layers in a streaming mode.

When an AL-Abort.request primitive is sent from the AL2 user to AL2 in order to abort a partially delivered AL-SDU, AL2 shall immediately send a MUX-Abort.request primitive to the MUX layer, if that AL-SDU has already been partially delivered to the MUX layer.

In the AL2 receiver, when a MUX-Abort.indication primitive is received from the MUX layer, any partially received AL-PDU should be discarded.

7.3.5 Procedures for sequence numbering

The following procedures apply when the SN field is in use.

Once a logical channel using AL2 is successfully opened according to the procedure defined in ITU-T H.245, the first AL-PDU transmitted by the AL2 sending entity shall have the SN field set to 0. For each subsequent transmitted AL-PDU which belongs to that logical channel, the value of the SN field shall be incremented by 1 modulo 256.

7.3.6 Procedures for error control

When the CRC check fails at the AL2 receiver, the associated AL-SDU may be delivered to the AL2 user, together with an appropriate error indication, via the AL-DATA.indication primitive.

When the SN field is in use, the AL2 receiver may detect that an AL-PDU is missing or has been misdelivered by the MUX layer. The AL2 receiver should discard any misdelivered AL-PDUs that it detects. For each missing AL-PDU that it detects, the AL2 receiver may deliver to the AL2 user an empty AL-SDU, together with an appropriate error indication, via the AL-DATA.indication primitive.

7.4 Adaptation Layer Type 3 (AL3) specification

7.4.1 Framework of AL3

AL3 is designed primarily for the transfer of digital video.

AL3 includes a 16-bit CRC for error-detection. AL3 also supports optional sequence numbering which may be used to detect missing and misdelivered AL-PDUs. AL3 transfers variable-length AL-SDUs and provides an optional retransmission procedure, designed primarily for video.

7.4.2 Primitives exchanged between AL3 and AL3 user

The information exchanged between AL3 and the AL3 user includes the following primitives:

- AL-DATA.request (AL-SDU).
- AL-DATA.indication (AL-SDU, EI).
- AL-Abort.request.
- AL-DRTX.indication.

7.4.2.1 Description of primitives

- AL-DATA.request: This primitive is issued by an AL3 user to AL3 to request the transfer of an AL-SDU to a corresponding AL3 user.
- AL-DATA.indication: This primitive is issued to an AL3 user by AL3 to indicate the arrival of an AL-SDU.
- AL-Abort.request: This primitive is issued to AL3 by an AL3 user to signal that a partially delivered AL-SDU is to be aborted.
- AL-DRTX.indication: This primitive is issued to an AL3 user by AL3 to indicate that a declined retransmission condition has occurred in the local transmitter.

7.4.2.2 Description of parameters

- AL-SDU: This parameter specifies the information exchanged between AL3 and the AL3 user. The length of the AL-SDU may be variable. Each transmitted AL-SDU shall contain an integral number of octets. The maximum size of AL-SDUs that an AL3 receiver can accept shall be signalled via the H.245 control channel.

  An AL3 receiving entity may deliver an empty AL-SDU to the AL3 user to indicate that an AL-SDU has been lost.

- Error Indication (EI): This parameter may be used by the AL3 receiver to pass error indications to the AL3 user. The precise procedures for using this parameter, and its numerical coding, are outside the scope of this Recommendation.

7.4.3 Functions, format and coding of AL3

7.4.3.1 Functions of AL3

AL3 provides the following functions:

- Error detection and indication.
- Optional sequence numbering.
- Optional support of retransmission.

7.4.3.2 Format and coding of AL3

The format of the AL-PDU is illustrated in Figure 7.

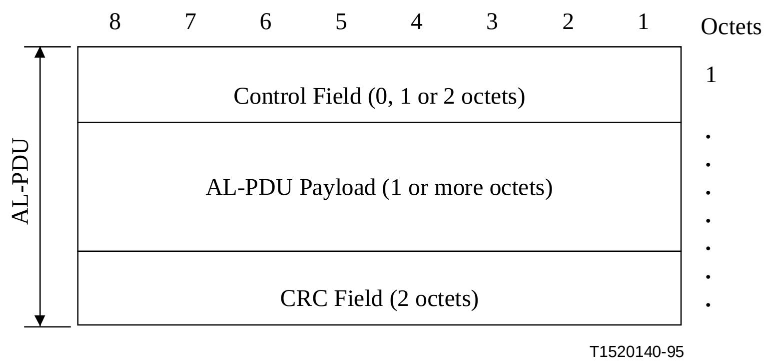
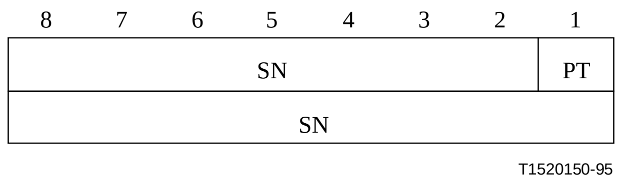


Figure 7/H.223 – AL-PDU format for AL3

7.4.3.2.1 Control field

The optional control field consists of a Payload Type (PT) field, which indicates the function of the AL-PDU payload, and a Sequence Number (SN) field, as shown in Figure 8.



PT Payload Type
SN Sequence Number

Figure 8/H.223 – Control field format of the AL-PDU for AL3

All receivers conforming to this Recommendation shall be capable of receiving and correctly interpreting AL-PDUs with 0-, 1- or 2-octet control fields. The actual number of octets in the control field is determined by the transmitter and shall be signalled to the far end in the OpenLogicalChannel message of ITU-T H.245.

When the control field is absent, the retransmission procedure is not used. However, the System Recommendation that uses ITU-T H.223 may require the control field to be present.

7.4.3.2.1.1 Payload Type (PT) field

The 1-bit PT field indicates the payload type of the AL-PDU. When the PT field is set to "1", the AL-PDU payload field shall contain an AL-SDU. Such an AL-PDU is referred to as an I-PDU. When the PT field is set to "0", the AL-PDU payload field shall contain a supervisory message used in the retransmission procedure. Such an AL-PDU is referred to as an S-PDU.

7.4.3.2.1.2 Sequence Number (SN) field

The sequence number field shall be 7 or 15 bits, depending on the length of the control field. In I-PDUs, the SN field shall contain a send sequence number N(S). In S-PDUs, the SN field shall contain a receive sequence number N(R) of an I-PDU as defined in 7.4.6.1.6.

Using the SN field, the AL3 receiver may detect that an AL-PDU is missing or has been misdelivered by the MUX layer.

The AL3 receiver should discard any misdelivered AL-PDUs that it detects.

7.4.3.2.2 AL-PDU payload field

The payload field of an I-PDU shall contain a complete AL-SDU received from the AL3 user where the first octet of the AL-PDU payload field shall be the first octet of the AL-SDU.

The 1-octet payload field of an S-PDU carries a supervisory message as defined in 7.4.6.2.

7.4.3.2.3 CRC field

The 16-bit CRC provides an error detection capability across the entire AL-PDU, including the control field, if used. The CRC and the CRC procedures are the same as those used in LAPM/V.42 and LAPF/Q.922.

The CRC has generator polynomial $g(x) = x^{16} + x^{12} + x^5 + 1$.

The CRC field shall be the one's complement of the sum (modulo 2) of

a) the remainder of $x^k (x^{15} + x^{14} + x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1)$ divided (modulo 2) by the generator polynomial $g(x) = x^{16} + x^{12} + x^5 + 1$, where k is the number of bits in the AL-PDU, not including the bits in the CRC field; and

b) the remainder of the division (modulo 2) by the generator polynomial $g(x) = x^{16} + x^{12} + x^5 + 1$, of the product of x^{16} multiplied by the content of the AL-PDU, excluding the bits in the CRC field. The polynomial representing the content of the AL-PDU is generated using bit number 1 of the first octet as the coefficient of the highest-order term.

As a typical implementation in the transmitter, the initial content of the register of the device computing the remainder of the division is preset to all 1's and is then modified by division by the generator polynomial (as described above) of the content of the AL-PDU, not including the bits in the CRC field; the one's complement of the resulting remainder is transmitted as the 16-bit CRC. The one's complement of the coefficient of the highest-order term of the remainder polynomial corresponds to bit number 1 of the first octet of the 16-bit CRC field. The one's complement of the coefficient of the lowest-order term of the remainder polynomial corresponds to bit number 8 of the second octet of the 16-bit CRC field.

NOTE – In contrast to the CRC procedure used for the 8-bit CRC in AL2, the CRC procedure used here includes pre- and post-conditioning.

7.4.4 Procedures for Abort

Abort procedures may be used when information is exchanged between layers in a streaming mode.

When an AL-Abort.request primitive is sent from the AL3 user to AL3 in order to abort a partially delivered AL-SDU, AL3 shall immediately send a MUX-Abort.request primitive to the MUX layer, if that AL-SDU has already been partially delivered to the MUX layer.

In the AL3 receiver, when a MUX-Abort.indication primitive is received from the MUX layer, any partially received AL-PDU should be discarded.

7.4.5 Procedures for Error Control

7.4.5.1 Invalid AL-PDUs

An invalid AL-PDU is one which:

a) has fewer than the minimum number of octets specified in 7.4.3.2, depending on the length of the control field; or

b) does not contain an integral number of octets; or

c) is longer than the maximum AL-PDU size; or

d) contains a CRC error.

An AL-PDU which is not invalid is referred to as a valid AL-PDU.

7.4.5.2 Error control control field absent

When the control field is absent, in case of a CRC failure at the AL3 receiver, the associated AL-SDU may be delivered to the AL3 user, together with an appropriate EI parameter, via the AL-DATA.indication primitive.

7.4.5.3 Error control control field present

When the control field is present, the AL3 receiver has the option of invoking the retransmission procedure. The sending AL3 entity shall respond to a retransmission request according to the procedures defined in 7.4.6.3.4. The error control procedures for retransmission are described in 7.4.6.

7.4.5.3.1 No retransmission

When the control field is in use and the AL3 receiver does not invoke the retransmission procedure, the following error control procedures may be used.

When the CRC check fails at the AL3 receiver, the associated AL-SDU may be delivered to the AL3 user, together with an appropriate EI parameter, via the AL-DATA.indication primitive.

Using the SN field, the AL3 receiver may detect that an AL-PDU is missing or has been misdelivered by the MUX layer.

The AL3 receiver should discard any misdelivered AL-PDUs that it detects.

For each missing AL-PDU that it detects, the AL3 receiver may deliver to the AL3 user an empty AL-SDU, together with an appropriate EI parameter, via the AL-DATA.indication primitive.

7.4.6 Retransmission procedure

The transmitter procedures defined in this clause shall be used when the control field is present. The receiver procedures defined in this clause shall be used when retransmission is used.

7.4.6.1 Definitions

7.4.6.1.1 Modulus

Each I-PDU, defined in 7.4.3.2.1.1, is sequentially numbered modulo 128 (2^{15}) and may have the value 0 through 127 (32767).

NOTE – All arithmetic operations on state variables and sequence numbers contained in this clause are modulo 128 (2^{15}).

7.4.6.1.2 Send state variable V(S)

V(S) is an internal variable of the transmitting AL3 entity. It denotes the sequence number of the next I-PDU to be transmitted. V(S) can take on the values 0 through 127 (32767). The value of V(S) shall be incremented by 1 after each in-sequence I-PDU is passed to the MUX layer in a MUX-SDU.

7.4.6.1.3 Send sequence number N(S)

Only I-PDUs contain N(S), the send sequence number of transmitted I-PDUs. At the time that an in-sequence I-PDU is designated for transmission, the value of N(S) is set equal to V(S).

7.4.6.1.4 Send buffer B_S

Each AL3 entity shall maintain a send buffer, B_S, used for storing the most recently transmitted I-PDUs. The minimum size of B_S that all AL3 transmitters must support is specified in the System Recommendation (e.g. ITU-T H.324) that uses ITU-T H.223. The actual size of B_S shall be indicated to the far end in the H.245 OpenLogicalChannel message.

7.4.6.1.5 Receive state variable V(R)

V(R) is an internal variable of the AL3 receiving entity. It denotes the sequence number of the next in-sequence I-PDU expected to be received. V(R) can take on the values 0 through 127 (32767). The value of V(R) shall be incremented by 1 with the receipt of a valid, in-sequence I-PDU whose N(S) equals V(R).

7.4.6.1.6 Receive sequence number N(R)

Only S-PDUs contain N(R), the send sequence number of an I-PDU that is referred to by the S-PDU.

7.4.6.2 Supervisory messages

S-PDUs carry supervisory messages. Each S-PDU contains one single-octet message. Table 3 shows the code assignment for the single-octet supervisory messages defined in AL3.

Table 3/H.223 – Code Assignment for supervisory messages

Supervisory Message	Code Number	Binary Code
SREJ	0	00000000
DRTX	255	11111111
Reserved	1-254	

7.4.6.2.1 Selective reject (SREJ) message

SREJ is used by an AL3 receiver to request the retransmission of the single I-PDU numbered N(R).

An SREJ PDU shall not be transmitted more than once for the same I-PDU.

7.4.6.2.2 Declined retransmission (DRTX) message

Since the error recovery procedures defined here only support negative acknowledgment, in certain conditions, the I-PDU(s) transmitted previously may have been discarded before the request for retransmission is received. The DRTX message is used by an AL3 transmitter to decline the requested retransmission of an I-PDU, when that I-PDU is not available in the send buffer at the time the SREJ PDU is received.

7.4.6.3 Detailed procedures

7.4.6.3.1 Initialization procedures

The retransmission procedures require that a reverse logical channel exist for sending supervisory messages.

Once the reverse logical channel has been established according to the procedure defined in ITU-T H.245, the AL3 entity shall:

– set V(S), V(R) to 0;

– clear any existing exception conditions.

7.4.6.3.2 Transmitting in-sequence I-PDUs

Information received from the AL3 user in an AL-SDU by means of an AL-DATA.request primitive shall be passed to the MUX layer in an I-PDU using the frame structure defined in 7.4.3.2. The SN field of the I-PDU shall be assigned the value V(S). V(S) shall be incremented by 1 after the I-PDU has been passed to the MUX layer.

7.4.6.3.3 Receiving in-sequence I-PDUs

When an AL3 entity receives a valid I-PDU, whose N(S) is equal to the current V(R), the AL3 entity shall increment its V(R) by 1.

7.4.6.3.4 Receiving SREJ PDUs

On receipt of a valid SREJ PDU, the AL3 entity shall act as follows:

a) if the I-PDU whose N(S) is equal to the N(R) of the SREJ PDU is still in the send buffer, the AL3 entity shall pass the corresponding I-PDU to the MUX layer as soon as possible. No other previously transmitted I-PDUs shall be retransmitted as a result of receiving the SREJ PDU.

b) if the I-PDU whose N(S) is equal to the N(R) of the SREJ PDU has been previously discarded, the AL3 entity shall enter a declined-retransmission exception condition. The procedures for this exception condition are defined in 7.4.6.4.5.

7.4.6.4 Exception condition reporting and recovery

Exception conditions may occur as a result of errors on the physical connection or procedural errors by an AL3 entity.

The error-recovery procedures that are available following the detection of an exception condition by an AL3 entity are defined in this clause.

7.4.6.4.1 Receiving invalid AL-PDUs

When a received AL-PDU is invalid, it is either discarded or saved for possible delivery later to the AL3 user.

7.4.6.4.2 N(S) sequence error

When there are no other outstanding exception conditions, an N(S) sequence error exception condition occurs in the receiving AL3 entity when a valid I-PDU is received containing an N(S) value that is not equal to the V(R) at the receiver. In this case, V(R) shall not be incremented, and one or more SREJ PDUs, each containing a different N(R), may be transmitted by the AL3 receiving entity to initiate an exception condition recovery for each SREJ PDU. After passing each SREJ PDU to the MUX layer, the AL3 entity shall start a local timer. Several factors that affect the length of the timer are given in Appendix IV/V.42. A different timer is maintained for each outstanding SREJ PDU. Successive SREJ PDUs are transmitted in the order indicated by their N(R) field.

For each SREJ PDU that it transmits, the AL3 receiver may pass an empty AL-SDU or an invalid received AL-SDU (previously saved), with an appropriate EI parameter, to the AL3 user via the AL-DATA.indication primitive.

When the retransmitted I-PDU with N(S) = V(R) is received, the exception condition for that I-PDU shall be cleared. The AL3 receiver should pass the associated AL-SDU, together with an appropriate EI parameter, to the AL3 user via the AL-DATA.indication primitive. When the exception condition is cleared, the associated timer shall be stopped and V(R) shall be incremented as many times as necessary so that V(R) represents the send sequence number of the next expected in-sequence I-PDU.

When a retransmitted I-PDU with N(S) ≠ V(R) is received, the AL3 receiver shall clear all exception conditions for any SREJ PDUs that may have been sent before the SREJ PDU for which the retransmission is received, by stopping the associated timers. For each exception condition cleared, the AL3 receiver shall increment V(R) by 1, and may deliver an empty AL-SDU, together with an appropriate EI parameter, to the AL3 user via the AL-DATA.indication primitive, prior to delivering the AL-SDU associated with the received I-PDU.

The information in all other received valid I-PDUs should be delivered to the AL3 user in AL-SDUs, together with an appropriate EI parameter.

7.4.6.4.3 N(R) sequence error

An N(R) sequence error exception condition occurs, when a valid S-PDU is received that contains an invalid N(R) value. An invalid N(R) value will result, when a first SREJ PDU is received with sequence number N(R) = N1, and then another SREJ PDU is received with N(R) = N2 and [V(S) – N2] is greater than or equal to [V(S) – N1].

An invalid N(R) may also result, when the N(R) value in a DRTX PDU is not equal to the N(R) value in an outstanding SREJ PDU.

The AL3 entity should ignore the message in such S-PDUs.

7.4.6.4.4 Procedure on expiration of timer

If the timer expires, the associated exception condition shall be cleared by stopping the timer and incrementing V(R). The AL3 receiver may pass an empty AL-SDU or an invalid received AL-SDU (previously saved), with an appropriate error indication, to the AL3 user via the AL-DATA.indication primitive.

7.4.6.4.5 Declined-retransmission condition

7.4.6.4.5.1 Error recovery procedures at the AL3 transmitter

Upon receiving an SREJ retransmission request, when the AL3 transmitter does not have the requested I-PDU stored in the send buffer, it shall:

– send a declined-retransmission (DRTX) PDU, whose N(R) value is equal to the N(R) value of the received SREJ PDU as soon as possible;

– send an AL-DRTX.indication to the AL3 user;
– resume transmission of AL-PDUs not yet transmitted.

7.4.6.4.5.2 Error recovery procedures at the AL3 receiver

When a DRTX message is received, the associated exception condition should be cleared by stopping the timer and incrementing V(R). The AL3 receiver may pass an empty AL-SDU or an invalid received AL-SDU (previously saved), with an appropriate error indication, to the AL3 user via the AL-DATA.indication primitive.

7.4.6.4.6 Unsolicited supervisory PDUs

An unsolicited DRTX PDU received by the AL3 entity should be ignored.

Any supervisory PDUs whose message code is reserved should be ignored.

ANNEX A

Multiplexing protocol for low bit rate multimedia mobile communication over low error-prone channels

A.1 General

This annex specifies the level 1 protocol of the mobile H.223 extensions as described in Annex C/H.324. This annex only changes MUX-PDU framing of the multiplex layer; however, the adaptation layer of ITU-T H.223 stays unchanged.

A.2 Multiplex (MUX) layer specification

The MUX-PDU framing of ITU-T H.223 is changed. Instead of 6.3, H.223 level 1 shall use the procedure in A.2.1.

A.2.1 MUX-PDU framing

In the basic mode, which is mandatory, all MUX-PDUs shall be delimited using 16-bit flags. Level 1 transmissions shall start in this basic mode.

In double-flag mode, which is optional, all MUX-PDUs shall be delimited by two consecutive 16-bit flags. All transmitters conforming to ITU-T H.223 shall signal their capability to delimit MUX-PDUs with two consecutive flags using the capability indication **h223AnnexADoubleFlag**, specified in ITU-T H.245.

A.2.1.1 Recommendation Flag

In the basic mode, all MUX-PDUs shall be preceded and followed by the 16-bit flag consisting of the following unique bit pattern (see Figure A.1):

8	7	6	5	4	3	2	1	Octets
1	1	1	0	0	0	0	1	1
0	1	0	0	1	1	0	1	2

Figure A.1/H.223 – 16-bit flag

The flag preceding the MUX-PDU is defined as the opening flag. The flag following the MUX-PDU is defined as the closing flag. The closing flag may also serve as the opening flag of the next MUX-PDU. However, all receivers conforming to ITU-T H.223 shall accommodate receipt of more than one consecutive flag, as the flag may be transmitted repetitively between MUX PDUs.

In the double-flag mode all MUX-PDUs shall be preceded and followed by two consecutive 16-bit flags ("double-flag"). The double-flag preceding the MUX-PDU is defined as the opening flag. The double-flag following the MUX-PDU is defined as the closing flag. The closing flag may also serve as the opening flag of the next MUX-PDU. All receivers working in the double-flag mode shall accommodate receipt of more than one consecutive double-flag, as the double-flag may be transmitted repetitively between MUX-PDUs.

A transmitter working in double-flag mode shall always transmit an even number of the 16-bit flags specified in Figure A.1.

If a transmitter has signalled MultiplexDoubleFlag capability, it shall start delimiting the MUX-PDUs with double-flags when it receives the **h223MultiplexReconfiguration. h223AnnexADoubleFlag.start** command.

It shall stop delimiting MUX-PDUs with double-flags when it receives the **h223MultiplexReconfiguration.h223AnnexADoubleFlag.stop** command.

In the period between requesting the change from basic mode to double-flag mode or vice versa and receiving the first flag of the new mode, the receiver should search for both single and double-flags. The first detected new flag should only be accepted as a valid flag if it is followed by a multiplex header with valid HEC. In order to increase the robustness of the change this search should be repeated multiple times until the new mode is established.

NOTE – Since Annexes B and C use an enhanced synchronization strategy, the delimiting of MUX-PDUs with double-flags is only performed in this annex.

A.2.1.2 Flag detection

The detection of the start of a MUX-PDU by the receiver may be done by correlation of the incoming bitstream with the synchronization flag. The output of the correlator may be compared with a Correlation Threshold (CT). The value of CT is not specified in this annex. Whenever the output is equal or greater than the threshold, the receiver should decide that a flag has been detected.

The octet aligned structure of the MUX-PDUs should be used to reduce the emulation of flags. Emulation may further be reduced by using the HEC check of the multiplex header.

NOTE – The level 1 procedure uses no 0-insertion method as described for HDLC flags in ITU-T H.223. This level does not prevent flag emulation in the bitstream and does not guarantee transparency.

ANNEX B

Multiplexing protocol for low bit rate multimedia mobile communication over moderate error-prone channel

B.1 General

This annex specifies the level 2 protocol of the mobile H.223 extensions as described in Annex C/H.324. This annex only changes MUX-PDU framing of the multiplex layer; however, the adaptation layer of ITU-T H.223 stays unchanged.

B.2 Abbreviations

For the purpose of this annex, the following abbreviation is added to clause 4.

MPL Multiplex Payload Length

B.3 Multiplex (MUX) layer specification

The MUX-PDU framing of ITU-T H.223 is changed. Instead of using 6.3 to 6.6, Level 2 shall use the following procedures and definitions.

B.3.1 MUX-PDU framing

See A.2.1.1 for basic mode. Consecutive synchronization flags shall not be used in level 2. This annex shall also not support double-flag mode. If the transmitter has no information to send, the stuffing-mode procedure of B.3.2.3 shall be used.

B.3.1.1 Flag detection

One fundamental property of the synchronization flag used in this annex is that it has an auto-correlation offering good detection properties, and the cross correlation between the flag and its one's complement exhibits the same strong detection property in the negative direction. Given a correlator which performs a synchronization search at specified positions, the output of that correlator can be used to signal additional information by detecting a one's complement flag for specific information. This is used within ITU-T H.223 for signalling the PM information, and for signalling transitions between levels.

The detection of the start of a MUX-PDU by the receiver should be done by correlation of the incoming bitstream with the MUX-PDU flag described in this clause. In determining the correlation sum, the correlator should interpret the zeros of the MUX-PDU flag to be "–1". The output of the correlator should then be compared with both a Correlation Threshold (CT) and its negative (–CT). The receiver should decide that a flag has been detected when the output of the correlator is either equal to or greater than CT, or if the output is less than or equal to –CT. The value of CT is not specified in this annex, but is instead left to the discretion of the implementor. The octet-aligned structure of the MUX-PDUs should be used to reduce the emulation of sync flags.

B.3.2 MUX-PDU format and coding/decoding

All MUX-PDUs shall conform to the format shown in Figure B.1.

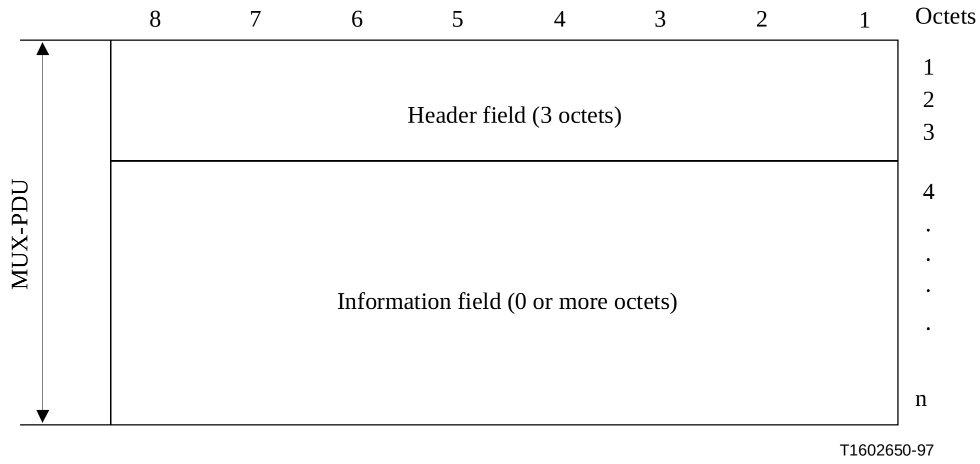


Figure B.1/H.223 – MUX-PDU format

B.3.2.1 Header field

The format of the header shall conform to the format shown in Figure B.2.

8	7	6	5	4	3	2	1	Octets
MPL4	MPL3	MPL2	MPL1	MC4	MC3	MC2	MC1	1
P4	P3	P2	P1	MPL8	MPL7	MPL6	MPL5	2
P12	P11	P10	P9	P8	P7	P6	P5	3

Figure B.2/H.223 – Header format of the MUX-PDU

The MC4 and MPL8 are the MSB of the MC and MPL fields, respectively. The P-bits are defined in B.3.2.1.3.

NOTE – The bit order of the fields in Figures B.2 and B.4 is not in accordance with the general convention of ITU-T H.223.

An optional header for this annex provides the capability to use the previous MUX-PDU whose header is corrupted due to channel errors. Figure B.3 shows the format of the MUX-PDU when using this option, and Figure B.4 shows the format of the optional header. The optional header contains the packet marker signalling and multiplex code of the previous header. This header belongs either to a non-stuffing MUX-PDU or a stuffing sequence.

The values of MC' and PM' are as indicated by MC and PM, respectively, in H.223/Level 0. The HEC' field shall be calculated from MC' according to the procedure described in 6.4.1.2. The use of this optional field shall be signalled by an H.245 "h223MultiplexReconfiguration.h223ModeChange.toLevel2withOptionalHeader" message, and started using the procedure defined in C.6/H.324.

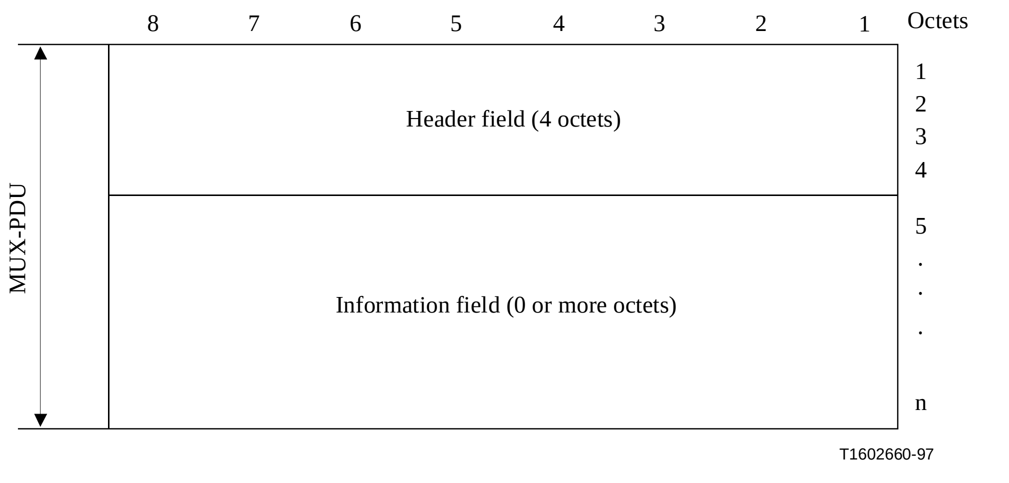


Figure B.3/H.223 – Optional MUX-PDU format

8	7	6	5	4	3	2	1	Octets
MPL4	MPL3	MPL2	MPL1	MC4	MC3	MC2	MC1	1
P4	P3	P2	P1	MPL8	MPL7	MPL6	MPL5	2
P12	P11	P10	P9	P8	P7	P6	P5	3
HEC' 3	HEC' 2	HEC' 1	MC' 4	MC' 3	MC' 2	MC' 1	PM'	4

Figure B.4/H.223 – Header format of the MUX-PDU

B.3.2.1.1 Multiplex Code (MC) field

See 6.4.1.1.

B.3.2.1.2 Multiplex Payload Length (MPL) field

The 8-bit MPL field describes the length of the information field in octets (see Figure B.2). The value of MPL shall be between 0 and 254. The value 255 shall not be used and is left for future use.

B.3.2.1.3 Parity bits field

The Extended Golay (24, 12, 8) code:

The Golay (23, 12, 7) code is a perfect code and in its conventional form shall be generated by the following generator polynomial:

$$G = 1 + X^2 + X^4 + X^5 + X^6 + X^{10} + X^{11}$$

The code shall be extended by adding an overall parity (even overall parity) check bit to produce a rate 1/2 code. The parity bits P shall be derived from the equation below.

$$\begin{bmatrix} P1 \\ P2 \\ P3 \\ P4 \\ P5 \\ P6 \\ P7 \\ P8 \\ P9 \\ P10 \\ P11 \\ P12 \end{bmatrix} = \begin{bmatrix} 1\,0\,1\,0\,1\,1\,1\,0\,0\,0\,1\,1 \\ 1\,1\,1\,1\,1\,0\,0\,1\,0\,0\,1\,0 \\ 1\,1\,0\,1\,0\,0\,1\,0\,1\,0\,1\,1 \\ 1\,1\,0\,0\,0\,1\,1\,1\,0\,1\,1\,0 \\ 1\,1\,0\,0\,1\,1\,0\,1\,1\,0\,0\,1 \\ 0\,1\,1\,0\,0\,1\,1\,0\,1\,1\,0\,1 \\ 0\,0\,1\,1\,0\,0\,1\,1\,0\,1\,1\,1 \\ 1\,0\,1\,1\,0\,1\,1\,1\,1\,0\,0\,0 \\ 0\,1\,0\,1\,1\,0\,1\,1\,1\,1\,0\,0 \\ 0\,0\,1\,0\,1\,1\,0\,1\,1\,1\,1\,0 \\ 1\,0\,1\,1\,1\,0\,0\,0\,1\,1\,0\,1 \\ 0\,1\,0\,1\,1\,1\,0\,0\,0\,1\,1\,1 \end{bmatrix}^{T} \bullet \begin{bmatrix} MC1 \\ MC2 \\ MC3 \\ MC4 \\ MPL1 \\ MPL2 \\ MPL3 \\ MPL4 \\ MPL5 \\ MPL6 \\ MPL7 \\ MPL8 \end{bmatrix}$$

NOTE – The symbol T denotes matrix transposition.

This code has a systematic structure. In the event of no channel errors, it is possible to extract the data without the need for complex decoding of the code word.

B.3.2.2 Information Field

See 6.4.2.

B.3.2.3 Stuffing mode

If no information is available, the stuffing mode shall be used. The multiplexer shall indicate a Level 2 stuffing mode by inserting a Level 2 synchronization flag followed by a Level 2 header (either the normal Level 2 header or the optional header in B.3.2.1, depending on the mode of operation). The MPL field shall be "00000000" and the MC shall be "0000". If the optional header is used in the stuffing operation mode, the optional header contains the packet marker signalling and multiplex code of the previous header. This header belongs either to a non-stuffing MUX-PDU or a stuffing sequence. This stuffing mode may be inserted consecutively an arbitrary number of times.

B.3.3 Marking of MUX-SDU boundaries

This clause replaces 6.5.

It is necessary to detect the boundaries of MUX-SDUs in the receiver in order to identify the location of all fields which the receiver must interpret in the AL and/or in a frame-oriented higher layer. This shall be accomplished as follows.

For non-segmentable logical channels, each MUX-SDU shall begin coincident with a slot specified in a single MultiplexElement structure whose type is logicalChannelNumber (see ITU-T H.245), and shall end after the specified repeatCount, or at the closing flag of the current MUX-PDU, whichever occurs first. The actual length of the MUX-SDU may be smaller than the length of the slot, provided that the current MUX-PDU is terminated by a closing flag immediately after the MUX-SDU. Since the size of each MUX-SDU may vary, multiple multiplex table entries may be defined to match the possible lengths of MUX-SDUs, in order to mix these MUX-SDUs with octets from other logical channels. It should be noted that the definitions given here together with the conditions given in ITU-T H.245 imply that it is allowed to place more than one MUX-SDU from a non-segmentable logical channel in one MUX-PDU, but only when the remote receiver has indicated the enhanced multiplex capability.

For segmentable logical channels, each MUX-SDU may be broken into segments and these segments may be transferred in one or more MUX-PDUs. A one's complemented closing flag shall be used to indicate that the last octet of the previous MUX-PDU was the final octet of the terminating MUX-SDU. As a result of this procedure, only one segmentable MUX-SDU is permitted to terminate within a MUX-PDU; as soon as the end of any MUX-SDU from a segmentable logical channel is reached, the MUX-PDU shall be terminated with a one's complemented closing flag. In all other circumstances the one's complemented flag shall not be used. Another result of this procedure is that a MUX-PDU will never contain octets from two different MUX-SDUs of the same segmentable logical channel.

ANNEX C

Multiplexing protocol for low bit rate multimedia mobile communication over highly error-prone channels

C.1 General

This annex specifies the level 3 protocol of the mobile H.223 extensions as described in Annex C/H.324. Level 3 defines the most error robust scheme of the mobile extensions of ITU-T-H.324. This annex changes both the multiplex layer and the adaptation layer of ITU-T H.223.

C.2 Acronyms

This annex uses the following acronyms:

ARQ	Automatic Repeat Request
CEC	Control Error Code
CF	Control Header Field
EGolay	Extended Golay (code)
FEC	Forward Error Correction
N(R)	Receive Sequence Number
N(S)	Send Sequence Number
RCPC	Rate Compatible Punctured Convolutional (code)

RN	Retransmission Number
SEBCH	Systematic Extended Bose-Chaudhuri-Hocquenghem (Code)
SN	Sequence Number
SRC	Systematic Recursive Convolutional (code)
TB	Tail Bit

C.3 Multiplex (MUX) layer specification

This annex uses nearly the same multiplex layer specifications as defined in Annex B, except for the stuffing mode of B.3.2.3.

C.3.1 Stuffing mode

Initially, prior to a dynamic level change, Level 3 stuffing mode shall have a structure identical to the stuffing mode used in Level 2 with MPL field set to "0000 0000". However, the MC field shall be set to "1111". The header may also include the optional header field of Annex B (see B.3.2.1). The stuffing mode may be inserted consecutively an arbitrary number of times. After a level change the terminal may also use the exact stuffing mode of B.3.2.3.

C.4 Adaptation Layer

C.4.1 AL1M

C.4.1.1 Framework of AL1M

AL1M is a highly flexible adaptation layer designed primarily for transfer of data and control information in highly error-prone environments such as could be encountered in a wireless environment. AL1M supports the use of error detection, Forward Error Correction (FEC), and retransmission (ARQ). AL1M also supports framed and unframed transfer mode.

AL1M provides two transfer modes:

a) framed transfer mode; and

b) unframed transfer mode.

In the framed transfer mode, AL1M may be used to transfer frames generated by a higher-layer protocol such as the data link layer protocol LAPM/V.42 or LAPF/Q.922. In this case, frames are first mapped to AL-SDUs and these are then passed by AL1M in MUX-SDUs to the MUX layer.

AL1M may also be used to carry an unframed octet sequence. In this mode, any internal framing present in the octet sequence is not visible to AL1M which passes the octets received from the higher layer to the MUX layer without paying any attention to framing.

The transfer mode of AL1M is selected by the transmitter in the H.245 OpenLogicalChannel message.

AL1M includes an optional EGolay or SEBCH coded header. AL1M also supports optional sequence numbering, which may be used to detect missing and misdelivered AL-PDUs. AL1M transfers variable-length AL-SDUs. In addition to ITU-T H.223, AL1M also supports the capability that a long framed AL-SDU may be split into several packets and delivered as one AL-SDU to the AL1-user.

C.4.1.2 Primitives exchanged between AL1M and AL1 user

The primitives exchanged are identical to those specified in 7.2.2, whereby the word AL1 shall be changed to AL1M.

C.4.1.2.1 Description of primitives

The description of primitives are identical to those specified in 7.2.2.1, whereby the word AL1 shall be changed to AL1M.

C.4.1.2.2 Description of parameters

- AL-SDU: This parameter specifies the unit of information exchanged between AL1M and the AL1 user. Each AL-SDU shall contain an integral number of octets. The length of AL-SDUs may be variable. The octets in an AL-SDU are numbered from 1 to n, and in each octet, bits are numbered from 1 to 8. Bit 1 of the octet 1 is transmitted first. An AL1M receiving entity may deliver an empty AL-SDU to the AL1 user to indicate that an AL-SDU is missing.
- Error Indication (EI): This parameter may be used in the AL1M receiver to pass error indications to the AL1 user. This may also be used if the AL1M receiver entity delivers an empty AL-SDU to the AL1 user. The precise procedures for using this parameter, and its numerical coding, are outside the scope of this Recommendation.

C.4.1.3 Functions of AL1M

AL1M provides the following functions:

- optional error detection and indication;
- optional sequence numbering;
- optional forward error correction;
- optional support of retransmission, either via ARQI or ARQII;
- optional AL-SDU splitting for framed frames.

C.4.1.4 Format and structure of AL1M

The format of the AL1M can be seen in Figure C.1.

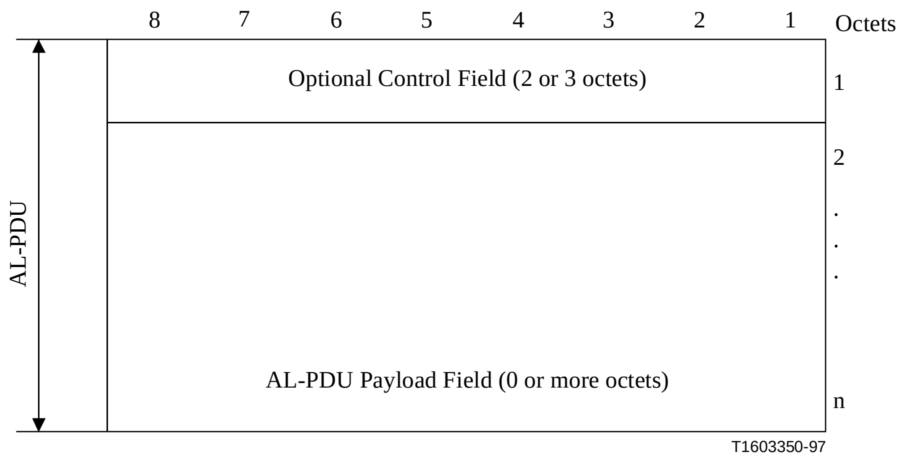


Figure C.1/H.223 – Format of the AL-PDU of the AL1M

The AL-PDU payload shall consist of either an I-PDU or an S-PDU. If an S-PDU is transmitted the length of the AL-PDU payload is 0, otherwise it is an I-PDU. In the following descriptions the AL-PDU payload is related as an I-PDU, if no other explicit explanation is given. The maximum length of AL-PDUs that an AL1M receiver can accept shall be signalled via the H.245 capability exchange.

In contrast to AL1 of ITU-T H.223, the AL-SDU is not always directly mapped to the AL-PDU payload, (see Figure C.2). The application layer (AL1 user) transfers its data through AL-SDUs to the adaptation layer. The adaptation layer forms its own AL-SDUs* from the AL-SDUs. The length of the AL-PDU can be derived from the procedure given in C.4.1.7.1. The AL-PDU is formed by the AL-PDU payload and the optional Control Field (CF). Optional bit interleaving may be applied to the entire AL-PDU.

The error protocol allows the AL1M to operate the following two modes:

– FEC_ONLY: In this mode an AL-SDU* with mandatory tail bits (TB)[1] and CRC is RCPC encoded with a code rate $r \leq 1.0$. The resulting AL-PDU only consists of an AL-PDU payload field. Splitting mode is not supported.

– ARQ: If the mode is set to ARQ (either ARQI or ARQII), it is possible to request retransmissions. The mandatory error detection code (CRC) and the mandatory tail bits (TB) are added to the AL-SDU*. This new field is encoded with the convolutional code of mother rate $r = ¼$. The encoded data may be filled to a linear buffer[2] according to the puncturing rule. For filling the AL-PDU payload, the octets of the buffer may be read in linear order from this buffer. The first octet of this buffer shall be the first octet of the AL-PDU payload.

  When only ARQI is used, each (re)-transmission shall contain the same encoded data. Therefore, the AL-PDU of each retransmission of the same SN shall contain the identical number of octets.

  Using ARQII, each (re)-transmission can contain different buffer encoded data, which may lead to different lengths for (re)-transmitted AL-PDU payloads. The first transmitted AL-PDU payload shall contain the octets of the buffer by linear reading from the beginning of the buffer. Each retransmission shall transmit data by reading the buffer after the last read octet. If the reading procedure comes to the end of the buffer, the procedure proceeds reading from the beginning of the buffer.

---

[1] The tail bits are required due to the use of the error-correction scheme with convolutional codes. In this case, the TB field has the length of 4 bits.

[2] The buffer scheme is used only for achieving an easy description of the encoding/decoding system. Therefore, this is no description on how to implement the system.

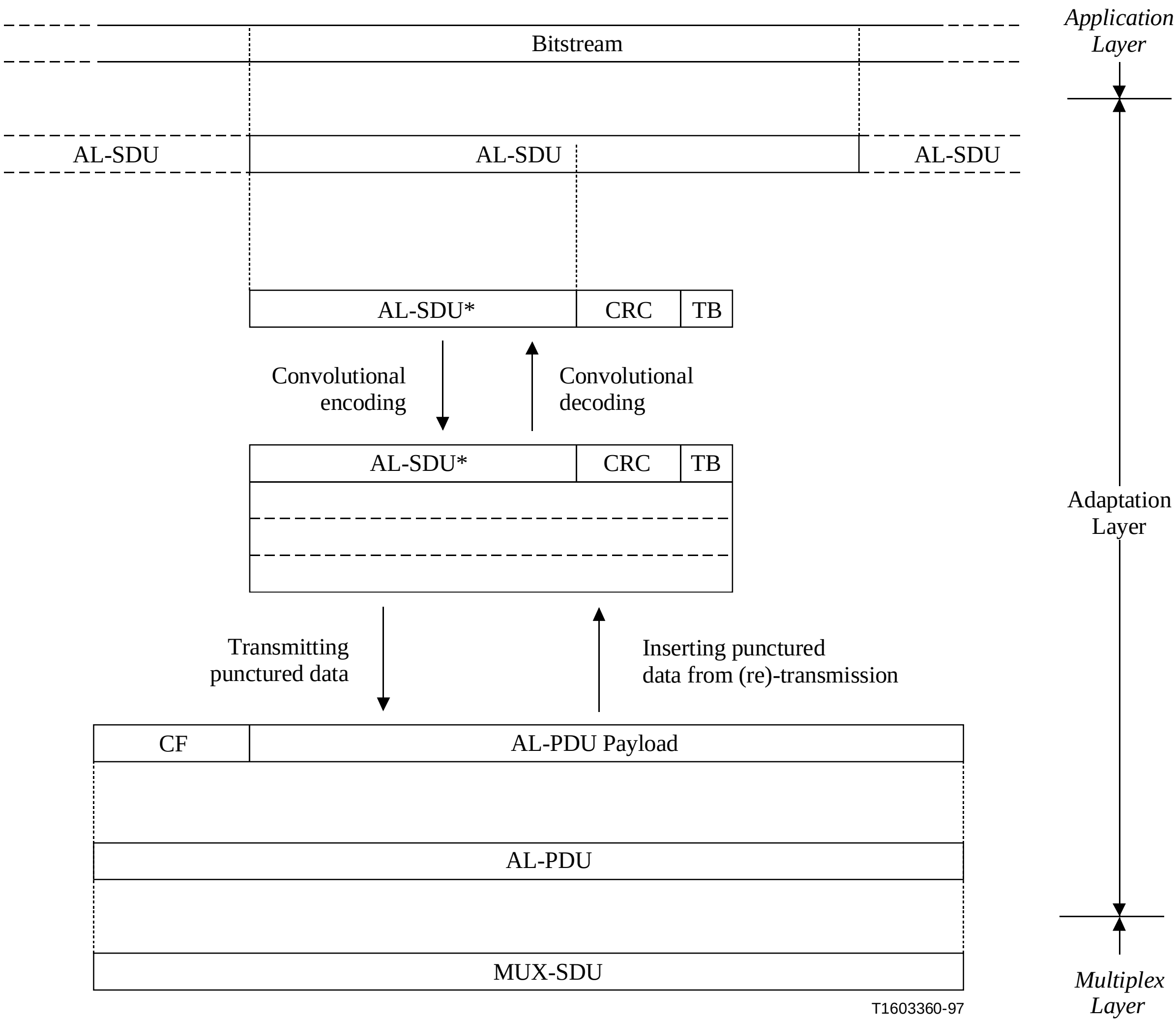


Figure C.2/H.223 – AL1M structure

C.4.1.5 Control Field (CF)

The optional control field consists of the Sequence Number (SN) field, the Retransmission Number (RN) field, the 1-bit field (X), and of the Control Error Code (CEC) field. The CEC uses either SEBCH or EGolay code, as illustrated in Figure C.3. These codes provide error detection and correction capability to the SN, RN and X fields.

8	7	6	5	4	3	2	1	Octets
P1	X	RN	SN5	SN4	SN3	SN2	SN1	1
P9	P8	P7	P6	P5	P4	P3	P2	2

Figure C.3/H.223 – Control field format of the AL-PDU for AL1M with SN = 5 and SEBCH code

8	7	6	5	4	3	2	1	Octets
SN8	SN7	SN6	SN5	SN4	SN3	SN2	SN1	1
P4	P3	P2	P1	X	RN	SN10	SN9	2
P12	P11	P10	P9	P8	P7	P6	P5	3

Figure C.4/H.223 – Control field format of the AL-PDU for AL1M with SN = 12 and EGolay code

NOTE – The bit order of the fields in Figures C.3 and C.4 is not in accordance with the general convention of ITU-T H.223.

Depending on the code used in the CEC field, the length of the SN field may vary, as indicated in Table C.1. When the control field is absent, the retransmission procedure is not used.

Table C.1/H.223 – Length of SN field according to different CECs

CEC	Length of SN field	Reference to
SEBCH(16, 7, 6)	5	Table I.2
EGolay(24, 12, 8)	10	B.3.2.1.3

C.4.1.5.1 Sequence Number (SN) field

The sequence number field shall be 5 or 10 bits, depending on the chosen CEC code. The SN field shall contain a send sequence number, N(S), except for in the case of SREJ messages. In this case it shall contain a receive sequence number, N(R).

When the SN field is in use, the AL1M receiver may detect that an AL-PDU is missing or has been misdelivered by the MUX layer. The AL1M receiver should discard any detected misdelivered AL-PDUs.

C.4.1.5.2 RN field

For an S-PDU in the reverse channel (SREJ message), the RN field shall contain the modulo 2 equivalent of the receive Retransmission Number (RN). Otherwise, this field is set to "0".

For an I-PDU frame, this field shall be used to signal the last transmitted packet resulting from the splitting of an AL-SDU into several AL-SDUs*. This shall only be done in a framed transfer mode. The splitting mode is described in C.4.1.6.

C.4.1.5.3 X field

For an S-PDU, the X field shall indicate either an SREJ or a DRTX message (see Table C.2).

Table C.2/H.223 – Definition of supervisory messages

Message	Bit value in X field
Selective reject (SREJ)	1
Declined retransmission (DRTX)	0

For an I-PDU, the X field shall be used as indication of the length of the AL-SDU* field. The X field shall be the modulo 2 equivalent of the number of octets within an AL-SDU*. If the AL-SDU* contains an odd number of octets, X = "1"; otherwise, X = "0".

C.4.1.5.4 Control Error Code (CEC) field

The CEC field defined by the parity bits P in Figures C.3 and C.4 provides error detection and/or error correction capability.

NOTE 1 – The bit order of the fields in Figures C.3 and C.4 is not in accordance with the general convention of ITU-T H.223.

Depending on the code used in the CEC field, the length of the SN field may vary, as indicated in Table C.1. When the control field is absent, the retransmission procedure is not used.

The definition of the EGolay code shall be the same as described in B.3.2.1.3, whereby the CEC shall be derived by the following equation:

$$
\begin{bmatrix} P1 \\ P2 \\ P3 \\ P4 \\ P5 \\ P6 \\ P7 \\ P8 \\ P9 \\ P10 \\ P11 \\ P12 \end{bmatrix} = \begin{bmatrix}
1&0&1&0&1&1&1&0&0&0&1&1 \\
1&1&1&1&1&0&0&1&0&0&1&0 \\
1&1&0&1&0&0&1&0&1&0&1&1 \\
1&1&0&0&0&1&1&1&0&1&1&0 \\
1&1&0&0&1&1&0&1&1&0&0&1 \\
0&1&1&0&0&1&1&0&1&1&0&1 \\
0&0&1&1&0&0&1&1&0&1&1&1 \\
1&0&1&1&0&1&1&1&1&0&0&0 \\
0&1&0&1&1&0&1&1&1&1&0&0 \\
0&0&1&0&1&1&0&1&1&1&1&0 \\
1&0&1&1&1&0&0&0&1&1&0&1 \\
0&1&0&1&1&1&0&0&0&1&1&1
\end{bmatrix}^{T} \bullet \begin{bmatrix} SN1 \\ SN2 \\ SN3 \\ SN4 \\ SN5 \\ SN6 \\ SN7 \\ SN8 \\ SN9 \\ SN10 \\ RN \\ X \end{bmatrix}
$$

NOTE 2 – The symbol T denotes matrix transposition.

The CEC bits of SEBCH code shall be derived by the following equation:

$$
\begin{bmatrix} P1 \\ P2 \\ P3 \\ P4 \\ P5 \\ P6 \\ P7 \\ P8 \\ P9 \end{bmatrix} = \begin{bmatrix}
1&0&0&0&1&0&1&1&1 \\
1&1&0&0&1&1&1&0&0 \\
0&1&1&0&0&1&1&1&0 \\
1&0&1&1&1&0&0&0&1 \\
0&1&0&1&1&1&0&0&1 \\
0&0&1&0&1&1&1&0&1 \\
0&0&0&1&0&1&1&1&1
\end{bmatrix}^{T} \bullet \begin{bmatrix} SN1 \\ SN2 \\ SN3 \\ SN4 \\ SN5 \\ RN \\ X \end{bmatrix}
$$

ALlM receivers conforming to ITU-T H.223 shall be capable of receiving and correctly interpreting AL-PDUs with these two different CECs. The actual CEC in the control field is determined by the transmitter and shall be signalled to the far end in the OpenLogicalChannel message of ITU-T H.245.

C.4.1.6 Procedures for splitting an AL-SDU (splitting mode)

Only in framed transfer the adaptation layer may split the AL-SDU into one or several AL-SDUs*, if the use of this splitting procedure is signalled by the OpenLogicalChannel message. This procedure is mandatory for the receiver.

Each AL-SDU* is transmitted as described in C.4.1.7. To identify the end of an AL-SDU, the last AL-SDU* of the AL-SDU shall be marked by setting the RN field to logical "1"; otherwise, the RN field shall be set to "0".

C.4.1.7 Procedures for encoding and decoding the AL-PDU payload

The payload field contains either a complete I-PDU or an S-PDU. The I-PDU is an octet-aligned field, which shall consist of one or more octets of encoded data. The S-PDU is only used if the retransmission capability was exchanged. The S-PDU is a 0-octet AL-PDU payload field. According to the direction of the S-PDU, either forward or reverse channel (see C.4.1.13.2) this 0-octet payload field represents different messages.

The transmitter shall construct an AL-PDU such that its size does not exceed the maximum AL-PDU size that the AL1M receiver can accept. This AL-PDU size is signalled in the H.245 capability exchange.

C.4.1.7.1 Evaluation of the AL-PDU (I-PDU) length

The following parameters are given:

- l_v Length of AL-PDU in bits;
- t Length of AL-SDU* in bits;
- r_{target} Code rate of the rate compatible punctured convolutional (RCPC) code;
- l_h Length of the control header (CF) field in bits;
- l_{CRC} Length of the cyclic redundancy check (CRC) field in bits;
- l_{TB} Number of tail bits of the RCPC code.

The length l_v of the AL-PDU can be evaluated by the following equation:

$$l_v = \min_{\lambda \in \Im, \lambda \bmod 8 = 0} \left\{ \lambda \geq l_h + \left\lceil \frac{(t + (l_{CRC} + l_{TB}))}{r_{target}} \right\rceil \right\}, \quad \text{with } \Im \text{ all integers} \tag{C-1}$$

The parameters l_v, t and $(l_{CRC}+l_{TB})$ shall be integer number of octets. However, equation (C-1) only guarantees that the resulting coding rate r_{result} is equal or smaller than the original rate r_{target} Equation (C-1) shall be used by the AL1M transmitter. At the AL1M receiver, the length of the AL-SDU* t shall be evaluated by the following equation:

$$t = \max_{\tau \in \Im, \bmod 8 = 0} \left\{ \tau \leq \lfloor (l_v - l_h) \rfloor \cdot r_{target} \right\} - 1_{CRC} - 1_{TB}, \quad \text{with } \Im \text{ all integers} \tag{C-2}$$

Both equations shall be calculated in octets, as illustrated by the following example:

Example:

The AL1M wants to transmit a AL-SDU* of t = 376 bits (47 octets), r_{target} = 8/10, l_h = 24 bits (3 octets), l_{CRC} = 20 bits, l_{TB} = 4 bits. Using the equation (C-1), the length of the AL-PDU is l_v = 66 octets. The parameter r_{result} can be evaluated by:

$$r_{result} = \frac{t + (l_{CRC} + l_{TB})}{l_v - l_h} \leq r_{target} \tag{C-3}$$

In this example:

$$r_{result} = \frac{50}{63} \approx 0.794 \leq r_{target} = 0.800$$

C.4.1.7.2 Cyclic Redundancy Check (CRC)

The CRC provides error detection capability. The CRC is appended to the AL-SDU* before the error correction coding procedure is done. The CRC is used by the AL1M receiver to verify whether the decoding procedure of the error correction algorithm is error-free. CRC lengths of 4, 12, 20 and 28 bits are supported. The length of the CRC field shall be specified during the H.245 OpenLogicalChannel procedure. The evaluation of the CRC shall be performed by the same procedure as described as in 7.3.3.2.3.

Description of the CRC polynomials:

a) 4-bit CRC: $x^4 + x^3 + x^2 + 1$;

b) 12-bit CRC: $x^{12} + x^{11} + x^3 + x^2 + x + 1$;

c) 20-bit CRC: $x^{20} + x^{19} + x^6 + x^5 + x^3 + 1$;

d) 28-bit CRC: $x^{28} + x^{27} + x^6 + x^5 + x^3 + 1$.

C.4.1.7.3 Systematic convolutional encoder

The channel encoder is based on a Systematic Recursive Convolutional (SRC) encoder with rate R = ¼. With the puncturing procedure described in Table C.4 we obtain a Rate Compatible Punctured Convolutional (RCPC) code. At the AL1M sending unit, the AL-PDU payload is generated by convolutional encoding of the concatenated field of the AL-SDU* and CRC field. The convolutional encoding of the CRC field starts with the highest term of the polynomial representing the CRC field. At the AL1M receiving entity, the concatenation of AL-SDU* and CRC field may be reconstructed by convolutional decoding, for example Viterbi decoding. As this code is systematic the receiver may also directly extract the CRC protected AL-SDU* from the received bitstream without convolutional decoding.

The SRC code is generated from a rational generator matrix by using a feedback loop. A shift register realization of the encoder is shown in Figure C.5.

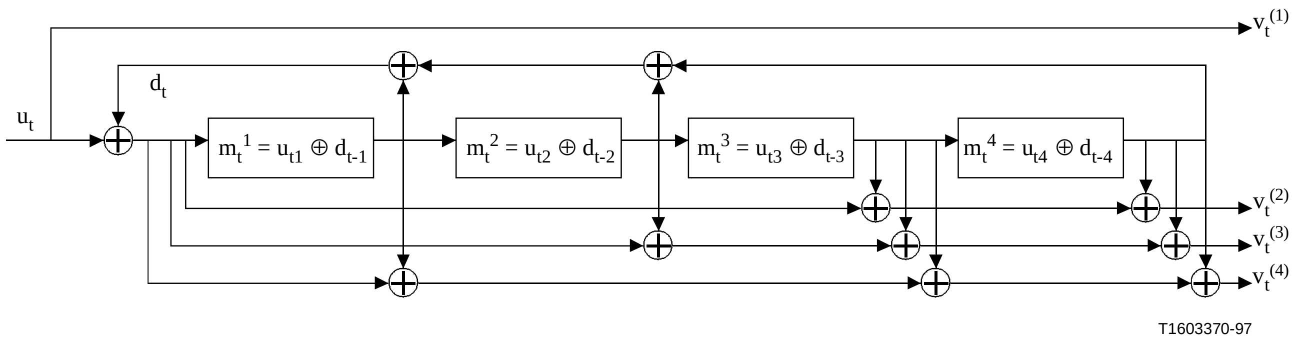


Figure C.5/H.223 – Shift register realization for systematic recursive convolutional encoder

To obtain the output vectors $\boldsymbol{v}_t$ at each time instant t, one has to know the content of the shift registers m_t^1, m_t^2, m_t^3, m_t^4, (corresponds to the state) and the input bit u_t at time t.

We obtain the outputs $v_t^{(2)}$, $v_t^{(3)}$ and $v_t^{(4)}$:

$$\boxed{\begin{array}{l} v_t^{(2)}, = m_t^4 \oplus m_t^3 \oplus (u_t \oplus d_t) \\ v_t^{(3)}, = m_t^4 \oplus m_t^3 \oplus m_t^2 \oplus (u_t \oplus d_t) \\ v_t^{(4)}, = m_t^4 \oplus m_t^3 \oplus m_t^1 \oplus (u_t \oplus d_t) \end{array}}$$

with:

$$d_t = m_t^4 \oplus m_t^2 \oplus m_t^1,\ m_t^4 = u_{t-4} \oplus d_{t-4},\ m_t^3 = u_{t-3} \oplus d_{t-3},\ m_t^2 = u_{t-2} \oplus d_{t-2},\ m_t^1 = u_{t-1} \oplus d_{t-1}$$

Finally we obtain for the output vector $\underline{\boldsymbol{v}}_t = (v_t^{(1)}, v_t^{(2)}, v_t^{(3)}, v_t^{(4)})$ at time t depending on the input bit u_t and the current state $\underline{\boldsymbol{m}}_t = (m_t^1, m_t^2, m_t^3, m_t^4)$:

$$\boxed{\begin{aligned}
v_t^{(1)} &= u_t \\
v_t^{(2)} &= m_t^4 \oplus m_t^3 \oplus (u_t \oplus d_t) = m_t^3 \oplus m_t^2 \oplus m_t^1 \oplus u_t \\
v_t^{(3)} &= m_t^4 \oplus m_t^3 \oplus m_t^2 \oplus (u_t \oplus d_t) = m_t^3 \oplus m_t^1 \oplus u_t \\
v_t^{(4)} &= m_t^4 \oplus m_t^3 \oplus m_t^1 \oplus (u_t \oplus d_t) = m_t^3 \oplus m_t^1 \oplus u_t
\end{aligned}}$$

with $\underline{\boldsymbol{m}}_1 = (m_1^1, m_1^2, m_1^3, m_t^4) = (0, 0, 0, 0) = \underline{\mathbf{0}}$

The initial state shall always be $\underline{\mathbf{0}}$, i.e. each memory cell contains a 0 before the input of the first information bit u_t. The tail bits following the information sequence $\boldsymbol{u}$ for returning to state $\underline{\boldsymbol{m}}_n = \underline{\mathbf{0}}$ (termination) depends on the last state $\underline{\boldsymbol{m}}_{n\text{-}3}$ (state after the input of the last information bit $u_{n\text{-}4}$). The termination sequence for each state described by $\underline{\boldsymbol{m}}_{n\text{-}3}$ is given in Table C.3. The receiver may use these Tail Bits (TB) for additional error detection.

The appendix ($u_{n\text{-}3}, u_{n\text{-}2}, u_{n\text{-}1}\ u_n$) to the information sequence can be calculated with the following condition:

For all t with $n\text{-}3 \leq t \leq n$: $u_t \oplus d_t = 0$.

Hence we obtain for the tail bit vector $\underline{\boldsymbol{u}}' = (u_{n\text{-}3}, u_{n\text{-}2}, u_{n\text{-}1}, u_n)$ depending on the state $\underline{\boldsymbol{m}}_{n\text{-}3} = (m_{n\text{-}3}^1, m_{n\text{-}3}^2, m_{n\text{-}3}^3, m_{n\text{-}3}^4)$.

$$\boxed{\begin{aligned}
u_{n\text{-}3} &= d_{n\text{-}3} = m_{n\text{-}3}^4 \oplus m_{n\text{-}3}^2 \oplus m_{n\text{-}3}^1 \\
u_{n\text{-}2} &= d_{n\text{-}2} = m_{n\text{-}2}^4 \oplus m_{n\text{-}2}^2 \oplus m_{n\text{-}2}^1 = m_{n\text{-}3}^3 \oplus m_{n\text{-}3}^1 \oplus 0 = m_{n\text{-}3}^3 \oplus m_{n\text{-}3}^1 \\
u_{n\text{-}1} &= d_{n\text{-}1} = m_{n\text{-}1}^4 \oplus m_{n\text{-}1}^3 \oplus m_{n\text{-}1}^2 = m_{n\text{-}3}^2 \oplus 0 \oplus 0 = m_{n\text{-}3}^2 \\
u_n &= d_n = m_{n\text{-}3}^1 \oplus 0 \oplus 0 = m_{n\text{-}3}^1
\end{aligned}}$$

Table C.3/H.223 – Tail bits for systematic recursive convolutional code

State $\underline{\boldsymbol{m}}_{n\text{-}3}$	$m_{n\text{-}3}^4$	$m_{n\text{-}3}^3$	$m_{n\text{-}3}^2$	$m_{n\text{-}3}^1$	$u_{n\text{-}3}$	$u_{n\text{-}2}$	$u_{n\text{-}1}$	u_n
0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	0	1
2	0	0	1	0	1	0	1	0
3	0	0	1	1	0	1	1	1
4	0	1	0	0	0	1	0	0
5	0	1	0	1	1	0	0	1
6	0	1	1	0	1	1	1	0
7	0	1	1	1	0	0	1	1
8	1	0	0	0	1	0	0	0
9	1	0	0	1	0	1	0	1
10	1	0	1	0	0	0	1	0

Table C.3/H.223 – Tail bits for systematic recursive convolutional code

State $\underline{m}_{n\text{-}3}$	$m_{n\text{-}3}^4$	$m_{n\text{-}3}^3$	$m_{n\text{-}3}^2$	$m_{n\text{-}3}^1$	$u_{n\text{-}3}$	$u_{n\text{-}2}$	$u_{n\text{-}1}$	u_n
11	1	0	1	1	1	1	1	1
12	1	1	0	0	1	1	0	0
13	1	1	0	1	0	0	0	1
14	1	1	1	0	0	1	1	0
15	1	1	1	1	1	0	1	1

C.4.1.7.4 Puncturing tables

Puncturing of the output of the SRC encoder allows different rates for transmission. The puncturing tables are listed in Table C.4. As all rates include all bits of all lower rates, this code is rate compatible.

Table C.4/H.223 – Puncturing tables (all values in hexadecimal representation)

Rate r	8/8	8/9	8/10	8/11	8/12	8/13	8/14	8/15	8/16	8/17	8/18	8/19	8/20
$P_r(0)$	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
$P_r(1)$	00	80	88	A8	AA	EA	EE	FE	FF	FF	FF	FF	FF
$P_r(2)$	00	00	00	00	00	00	00	00	00	80	88	A8	AA
$P_r(3)$	00	00	00	00	00	00	00	00	00	00	00	00	00

Rate r	8/21	8/22	8/23	8/24	8/25	8/26	8/27	8/28	8/29	8/30	8/31	8/32
$P_r(0)$	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
$P_r(1)$	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
$P_r(2)$	EA	EE	FE	FF	FF	FF	FF	FF	FF	FF	FF	FF
$P_r(3)$	00	00	00	00	80	88	A8	AA	EA	EE	FE	FF

C.4.1.8 Interleaving

For some channels, block interleaving may be used.

If interleaving is used, it shall be applied to the entire AL-PDU including the control field. As the length of the AL-PDU varies, the dimension of the block interleaver matrix has to be recalculated for each length. Given a AL-PDU of length l_v, the dimensions, the width a and the height b of the block interleaver can be calculated:

$$a = \max_{\alpha \in \Im, l_v \bmod \alpha = 0} \left\{ \alpha \le \sqrt{l_v} \right\} \text{ with } \Im \text{ all integers}$$

$$b = l_v / a$$

b represents the distance after interleaving between two bits that are adjacent to each other in the AL-PDU.

The receiver shall calculate the dimensions of the interleaver with the upper equation and the length of the received AL-PDU l_v. De-interleaving shall also be applied to the entire AL-PDU.

The process of block interleaving with the width a and the height b is as follows:

1) Prepare a rectangular buffer with a columns and b rows.
2) The input data is written in to the buffer from the top left to the bottom right, row by row, bit by bit.
3) The output data is read out from the buffer from the top left to the bottom right, column by column, bit by bit.

This is represented with a formula as follows:

x_i: i-th input bit to the interleaver. $i = 0 \dots N-1$,

y_j: j-th output bit from the interleaver. $j = 0 \dots N-1$,

$y_j = x_i$,where $i = (j \bmod b) \cdot a + [j/b]$.

N is the number of bits input to the interleaver, and $\lceil x \rceil$ is the maximum integer value which is smaller than or equal to x.

C.4.1.9 Encoding procedure: AL-SDU* (I-PDU) to AL-PDU

C.4.1.9.1 Encoding

The following steps are necessary to obtain an AL-PDU from an AL-SDU*.

1) Calculate the length of the AL-PDU l_v according to C.4.1.7.1 and the first rate required in the H.245 OpenLogicalChannel message.
2) The CRC of the length required in the H.245 OpenLogicalChannel message field shall be added.
3) Due to the convolutional codes of mother rate r = ¼ with memory 4, four Tail Bits (TB) shall be appended from Table C.3.
4) Generate the encoded data by passing them through the convolutional encoder.
5) According to the puncturing rules of Table C.4, fill the bits of the convolutional encoder output into a linear buffer. Put AL-SDU* with appended CRC and TB at the beginning of this buffer.
6) For the first transmission read $[l_v - l_h]$ bits (AL-PDU payload length) from the buffer, starting from the beginning of the buffer and fill these bits into the AL-PDU payload field. The first octet of the buffer is the first octet of the AL-PDU payload field.
7) The Control Field (CF) shall not be used if the ARQ mode, signalled by the H.245 message, is set to "noArq".
8) Interleaving according to C.4.1.8 shall be applied for the entire AL-PDU, if this has been required in the H.245 OpenLogicalChannel message.

These steps are valid for the modes FEC_ONLY, ARQI and ARQII. If FEC_ONLY is used, no retransmission is possible.

Using the ARQ mode, the content of the AL-PDU varies for the retransmissions:

– ARQI: In this mode, the content of each (re)-transmitted AL-PDU is the same and has the same length.

– ARQII: If $V^j(S) = 0$, the encoding procedure step 6 of this clause shall be performed. Otherwise, the transmitter may choose any AL-PDU payload length, whereby the AL-PDU payload length shall be an integral number of octets. This AL-PDU payload shall be read in consecutive order from the linear buffer.

  However if the mother code rate is reached, the transmitter begins transmitting at the beginning of the linear buffer and is still free to choose the code rate, if the maximum number of retransmissions is not reached.

Figure C.6 illustrates the encoding procedures of the ALM at the transmit side.

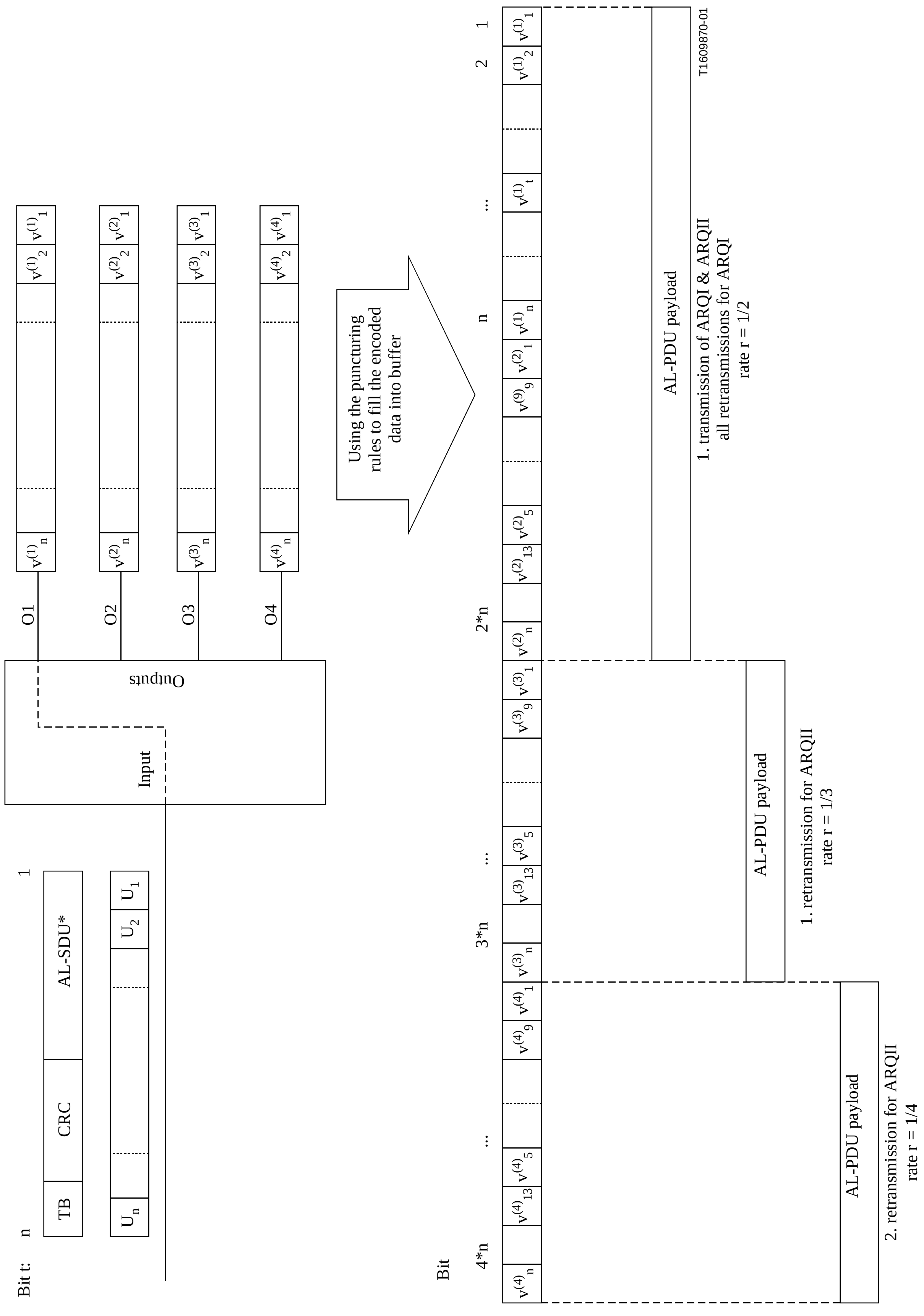


Figure C.6/H.223 – Encoding procedure of the AL3M at the transmitter side

C.4.1.9.2 Possible implementation of mapping procedure

The mapping from the temporary matrix to the linear buffer is done by the rules of the puncturing Table C.4 that describes the exact reading order from the temporary matrix. Table C.5 reflects that reading order for the output 2, 3 and 4.

Table C.5/H.223 – Reading order for the output 2, 3 and 4 of the temporary matrix of Figure C.7

column number	1	2	3	4	5	6	7	8
reading order	1	5	3	7	2	6	4	8

The linear buffer is filled in the following way:

1) The first output line of the convolutional encoder is directly written to the linear buffer.

2) The columns of output 2 of the temporary matrix are written to the linear buffer by the use of Table C.5. Then all the bits in column 1 are read from the top to bottom and filled to the linear buffer, followed by column 5 and so on. After having read all columns, the mapping procedure continues with output 3.

3) The columns of output 3 of the temporary matrix are written to the linear buffer by the use of Table C.5. Then all the bits in column 1 are read from the top to bottom and filled to the linear buffer, followed by column 5 and so on. After having read all columns, the mapping procedure continues with output 4.

4) The columns of output 4 of the temporary matrix are written to the linear buffer by the use of Table C.5. Then all the bits in column 1 are read from the top to bottom and filled to the linear buffer, followed by column 5 and so on. After having read all columns, the mapping procedure is finished.

C.4.1.9.3 Example

The following example demonstrates how to interpret the puncturing tables and how to fill the encoded data into the linear buffer.

The following parameters are given:

- l_{CRC} = 4 bits;
- l_{TB} = 4 bits;
- t = 8 bits;
- l_{buffer} = 4 × 16 bits = 64 bits.

The convolutional encoder gives four output streams. Each of these output streams consists of 16 bits. The numbering is equivalent to that of Figure C.6. The output of line 1, i.e. the systematic bits are directly transferred to the linear buffer. All bits of output 2, 3 and 4 are filled into a temporary matrix, which is used to give a simple example description. The bits are read line by line from the convolutional encoder output and are written from column 1 to 8 and then row by row into the temporary matrix, as shown Figure C.7. Then this matrix is mapped using the puncturing rule and appended to the linear buffer. The puncturing rule describes in which order the columns are read. To achieve a code rate of r = 1/3, the first 48 bits are transmitted to the AL-PDU payload by reading the bits from the start position of the buffer (right bottom in Figure C.6). Another code rate r = 8/13 requires that 26 bits be read from the matrix. However, 32 bits shall be transmitted to obtain an octet-aligned AL-PDU payload.

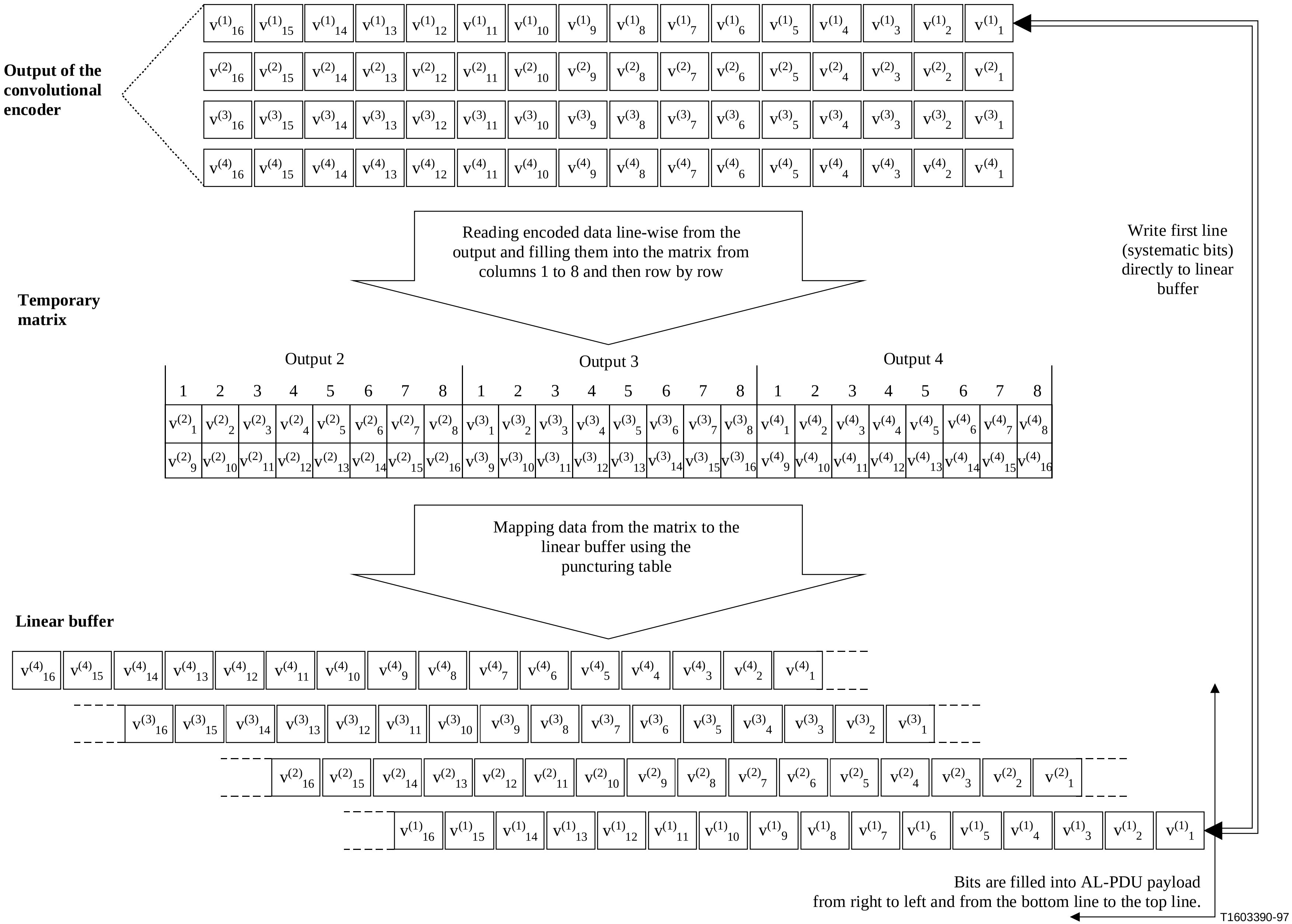


Figure C.7/H.223 – Ordering of the bit streams of the encoded data

C.4.1.10 Decoding of the AL-PDU payload (I-PDU)

The receiver may check the received systematic bits before decoding the convolutional code. It may also use the tail bits for error detecting. If CRC or TB check fails, any kind of convolutional decoding may be used.

After convolutional decoding, the CRC should be used to check the correctness of the decoding attempt again. If the CRC fails, another retransmission may be requested, or the wrong data may be given to the AL1 user with an appropriate error indication (EI) message. If only wrong data is available, the receiver may use the decoded information bits or the systematic bits before decoding as received AL-SDU*.

If ARQI retransmission procedure is used, each retransmission gives the same data as the previous one. If the ARQII procedure is applied, each retransmission is delivering new data which may be combined with the previous received data to achieve a more powerful error correction code. After each decoding attempt, the decoding result should be checked by the CRC.

C.4.1.11 Procedures for Abort

This primitive is discarded and no action is undertaken.

C.4.1.12 Procedures for error control

C.4.1.12.1 Invalid AL-PDUs

An invalid AL-PDU occurs in either of the following cases:

a) The associated AL-PDU has fewer than the minimum number of octets specified in C.4.1.4.

b) The AL-PDU does not contain an integral number of octets.

c) The AL-PDU is longer than the maximum AL-PDU size.

d) The AL-SDU* does not contain an integral number of octets.

An AL-PDU which is not invalid is referred to as a valid AL-PDU.

C.4.1.12.2 Errored AL-PDUs

An errored AL-PDU occurs at the AL1M receiver in the following case:

– The AL-PDU is valid and the associated error decoded AL-PDU Payload contains a CRC error.

An AL-PDU which is valid and not errored is referred to as an error-free AL-PDU.

C.4.1.12.3 Error control: CF absent

In case of a CRC failure at the AL1M receiver when the Control Field (CF) is absent and CRC error detection is used for the AL-PDU Payload, the associated AL-PDU Payload shall be delivered to the AL1 user, together with an appropriate EI parameter, via the AL-DATA.indication primitive.

C.4.1.12.4 Error control: Forward control field present

When the CF is present, the AL1M receiver has the option of invoking the retransmission procedures ARQI or ARQII. Which of these are used shall be indicated by the AL1M sending entity in the H.245 OpenLogicalChannel message. The AL1M sending entity shall respond to a retransmission request according to the procedures defined in C.4.1.13. The error control procedures for retransmission are described in C.4.1.13.8.

When the CF is in use and the AL1M receiver does not invoke the retransmission procedure, the procedure described in 7.4.5.3.1 should be followed by using AL1M instead of AL3.

C.4.1.13 Retransmission procedures (ARQI, ARQII)

This clause addresses the two retransmission procedures ARQI and ARQII. The transmitter procedures defined in this clause shall be used when the control field is present. The receiver procedures defined in this clause shall be used when retransmission is used.

C.4.1.13.1 Definitions

a) *Modulo*

Each AL-PDU Payload is sequentially numbered modulo 2^5 or 2^{10} and may have the value 0 through $2^5 - 1$ or $2^{10} - 1$. The length of the sequence number field (SN) is set with the OpenLogicalChannel message of ITU-T H.245.

NOTE – All arithmetic operations on state variables and sequence numbers contained in this subclause are modulo 2^5 or 2^{10}.

b) *Send sequence variable V(S)*

V(S) is an internal variable of the transmitting AL1M entity. It denotes the sequence number of the next AL-PDU Payload to be transmitted to the far end. V(S) can take on the values 0 through 2^5 or 2^{10}. The value of V(S) shall be incremented by 1 after each in-sequence AL-PDU is passed to the MUX layer in a MUX-SDU.

c) *Send retransmission variables $V^j(S)$*

$V^j(S)$ are internal variables of the transmitting AL1M entity. A separate counter $V^j(S)$ exists for each possible value j of V(S). $V^j(S)$ can take on the values 0 through R_{max}. The value of $V^j(S)$ shall be incremented by 1 after each (re-)transmission of an AL-PDU for an AL-SDU with sequence number j. The value of $V^j(S)$ shall be set to 0 in the following cases:

- at initialization;
- when the send buffer B_S no longer contains information for the corresponding AL-PDU payload.

d) *Send sequence number N(S)*

AL-PDUs contain N(S), the send sequence number of the corresponding AL-PDU Payloads. At the time that an in-sequence AL-PDU is designated for transmission, the value of N(S) is set equal to V(S).

e) *Maximum number of retransmissions R_{max}*

R_{max} is a parameter that indicates the maximum number of retransmissions allowed. Its value shall be indicated by the AL1M transmitting unit in the H.245 OpenLogicalChannel message.

f) *Send buffer B_S*

Each AL1M entity shall maintain a send buffer, B_S, used for storing the most recently transmitted AL-PDU Payload information. The minimum size of B_S that all AL1M transmitters shall support is specified in the System Recommendation (e.g. ITU-T H.324) that uses this annex. The actual size of B_S shall be indicated to the far end in the H.245 OpenLogicalChannel message.

g) *Receive sequence variable V(R)*

V(R) is an internal variable of the AL1M receiving entity. It denotes the sequence number of the next in-sequence AL-PDU expected to be received. V(R) can take on the values 0 through 2^5 or 2^{10} . The value of V(R) shall be incremented by 1 with the receipt of a valid, in-sequence AL-PDU whose N(S) equals V(R).

h) *Receive retransmission variables $V^j(R)$*

$V^j(R)$ are internal variables of the AL1M receiving entity. $V^j(R)$ can take values 0 through R_{max}. The value of a variable $V^j(R)$ shall be used to monitor the number of retransmissions requested. When the ARQII error protection scheme is used, the value of a variable $V^j(R)$ shall also be used to determine the number i of the next AL-PDU payload z_i to be received from the AL1M transmitting entity.

The value of a variable $V^j(R)$ shall be incremented by 1 with the receipt of an errored AL-PDU with N(S) = j.

The value of a variable $V^j(R)$ shall be set to 0 when the received AL-PDU with N(S) = j results in an error-free decoding of the corresponding AL-PDU Payload.

i) *Receive retransmission number RN*

Only the header field of the reverse channel contains RN, the receive retransmission number. When requesting a retransmission, this 1-bit number shall be set to the parity of the receive retransmission variable of the requested AL-PDU payload.

j) *Receive sequence number N(R)*

Only the header field of the reverse channel contains N(R), the receive sequence number of an AL-PDU that is referred to by the reverse header field.

C.4.1.13.2 Supervisory messages

According to the direction, either forward or reverse channel, an S-PDU with different messages is transmitted:

- an S-PDU from the transmitter to receiver (forward channel) signals a *DRTX* message;
- an S-PDU from the receiver to the transmitter (reverse channel) carries an *SREJ* message.

Selective reject (SREJ) message

SREJ is used by an AL1M receiver to request a retransmission for the single AL-PDU Payload numbered N(R). An SREJ message shall not be transmitted more times than the negotiated maximum number of retransmissions R_{max} for the same AL-PDU Payload.

Declined retransmission (DRTX) message

Since the error recovery procedures defined here only support negative acknowledgment, in certain conditions, the AL-PDU Payload information transmitted previously may have been discarded before the request for retransmission is received. The DRTX message is used by an AL1M transmitter to decline the requested retransmission for an AL-PDU Payload, when the information for that AL-PDU Payload is not available in the send buffer at the time the SREJ message is received.

C.4.1.13.3 Initialization procedures

The retransmission procedures require that a reverse logical channel exist for sending supervisory messages.

Once the reverse logical channel has been established according to the procedure defined in ITU-T H.245, the AL1M entity shall:

- set V(S), V(R), $V^j(S)$, $V^j(R)$ to 0;
- clear any existing exception conditions.

C.4.1.13.4 Transmitting in-sequence I-PDUs

Information received from the AL1 user in an AL-SDU by means of an AL-DATA.request primitive shall be passed to the MUX layer in an I-PDU using the frame structure defined in C.4.1.4. The SN field of the I-PDU shall be assigned the value V(S). V(S) shall be incremented by 1 after the I-PDU has been passed to the MUX layer.

C.4.1.13.5 Receiving in-sequence I-PDUs

When an AL1M entity receives a valid I-PDU whose N(S) is equal to the current V(R), the AL1M entity shall increment its V(R) by 1.

C.4.1.13.6 Receiving SREJ-PDUs

On receipt of a valid SREJ-PDU, the AL1M entity shall act as follows:

a) If the I-PDU, whose N(S) is equal to the N(R) of the SREJ message is still in the send buffer, the AL1M entity shall pass a corresponding AL-PDU to the MUX layer as soon as possible.

   When ARQI error protection is used, the same AL-PDU payload shall be used for re-transmission.

   When ARQII is used, the parity of the send retransmission variable $V^j(S)$ is checked against the 1-bit receive retransmission number RN. If the parity differs, $V^j(S)$ will be decremented by 1. Then the next I-PDU payload, according to the procedure described in C.4.1.9, shall be retransmitted to the receiver.

   No other previously transmitted I-PDUs shall be retransmitted as a result of receiving the SREJ-PDU.

b) If the AL-PDU whose N(S) is equal to the N(R) of the SREJ message has been previously discarded, the AL1M entity shall enter a declined-retransmission exception condition. The procedures for this exception condition are defined in C.4.1.13.8 e).

C.4.1.13.7 Transmitting SREJ messages

When a valid but errored I-PDU is received and $V^j(R) < R_{max}$, a SREJ message shall be produced with the receive sequence number N(R) set to the N(S) from the errored I-PDU and the modulo 2 of $V^j(R)$ set to RN field. The corresponding receive retransmission variable $V^j(R)$ shall be incremented.

C.4.1.13.8 Exception condition reporting and recovery

Exception conditions may occur as a result of errors on the physical connection or procedural errors by an AL1M entity.

The error-recovery procedures that are available following the detection of an exception condition by an AL1M entity are defined in this clause.

a) *Receiving invalid AL-PDUs*

   When a received AL-PDU is invalid, it is either discarded or saved for possible delivery later to the AL1 user.

b) *N(S) sequence error*

   When there are no other outstanding exception conditions, an N(S) sequence error exception condition occurs in the receiving AL1M entity when a valid I-PDU is received containing an N(S) value that is not equal to the V(R) at the receiver. In this case, V(R) shall not be incremented, and one or more SREJ-PDUs, each containing a different N(R), may be transmitted by the AL1M receiving entity to initiate an exception condition recovery for each SREJ-PDU. After passing each SREJ-PDU to the MUX layer, the AL1M entity shall start a local timer. Several factors that affect the length of the timer are given in

Appendix IV/V.42. A different timer is maintained for each outstanding SREJ-PDU. Successive SREJ-PDUs are transmitted in the order indicated by their N(R) field.

For each SREJ-PDU that it transmits, the AL1M receiver may pass an empty AL-SDU or an invalid received AL-SDU (previously saved), with an appropriate EI parameter, to the AL1 user via the AL-DATA.indication primitive.

When the retransmitted I-PDU with N(S) = V(R) is received, the exception condition for that I-PDU shall be cleared. The AL1M receiver should pass the associated AL-SDU, together with an appropriate EI parameter, to the AL1 user via the AL-DATA.indication primitive. When the exception condition is cleared, the associated timer shall be stopped and V(R) shall be incremented as many times as necessary so that V(R) represents the send sequence number of the next expected in-sequence I-PDU.

When a retransmitted I-PDU with N(S) ≠ V(R) is received, the AL1M receiving unit clears all exception conditions related to previously sent SREJ-PDUs for which retransmission is received, by stopping the associated timers. For each exception condition cleared, the AL1M receiver shall increment V(R) by 1, and may deliver an empty AL-SDU, together with an appropriate EI parameter, to the AL1 user via the AL-DATA.indication primitive, prior to delivering the AL-SDU associated with the received I-PDU.

The information in all other received valid I-PDUs should be delivered to the AL1 user in AL-SDUs, together with an appropriate EI parameter.

c) *N(R) sequence error*

An N(R) sequence error exception condition occurs when a valid S-PDU with an invalid N(R) value is received. An invalid N(R) value will be produced when a first SREJ-PDU received with sequence number N(R) = N1 is followed by another SREJ-PDU with N(R) = N2, and (V(S) – N2) is greater than, or equal to, (V(S) – N1).

An invalid N(R) may also occur when the N(R) value in a DRTX-PDU is not equal to the N(R) value in an outstanding SREJ-PDU.

The AL1M entity should ignore the message in such S-PDUs.

d) *Procedure on expiration of timer*

If the timer expires, the associated exception condition shall be cleared by stopping the timer and incrementing V(R). The AL1M receiver may pass an empty AL-SDU or an invalid received AL-SDU (previously saved), with an appropriate error indication, to the AL1 user via the AL-DATA.indication primitive.

e) *Declined-retransmission condition*

Error recovery procedures at the AL1M sending unit

Upon receiving an SREJ retransmission request, when the AL1M transmitter does not have the information for the requested AL-PDU Payload stored in the send buffer, it shall:

- send a declined-retransmission (DRTX) message whose N(R) value is equal to the N(R) value of the received SREJ message as soon as possible;
- send an AL-DRTX.indication to the *AL1 user;*
- resume transmission of AL-PDUs not yet transmitted.

Error recovery procedures at the AL1M receiving unit

When a DRTX message is received, the associated exception condition should be cleared by stopping the timer and incrementing V(R). The AL1M receiver may pass an empty AL-SDU or an invalid received AL-SDU (previously saved), with an appropriate error indication, to the AL1 user via the AL-DATA.indication primitive.

C.4.1.13.9 Unsolicited supervisory PDUs

An unsolicited DRTX-PDU received by the AL1M entity should be ignored.

C.4.2 AL2M

C.4.2.1 Framework of AL2M

AL2M is primarily designed for the transfer of digital audio in highly error-prone channels.

AL2M only provides optional sequence numbering and optional AL-PDU interleaving. Therefore, any additional error control may be provided by the higher layer protocol. For example, the Annex C/G.723.1 defines such an error control procedure.

The AL-SDU and the AL-PDU shall be octet aligned.

Audio frames are first mapped to AL-SDU and these are then passed by AL2M in MUX-SDUs with an optional AL2M-header and optional interleaving to the MUX layer.

C.4.2.2 Primitives exchanged between AL2M and AL2 user

The information exchanged between AL2M and an AL2 user includes the following primitives:

– AL-DATA.request (AL-SDU);

– AL-DATA.indication (AL-SDU, EI);

– AL-Abort.request.

C.4.2.2.1 Description of primitives

– AL-DATA.request: This primitive is issued by an AL2 user to AL2M to request the transfer of an AL-SDU to a corresponding AL2 user.

– AL-DATA.indication: This primitive is issued to an AL2 user by AL2M to indicate the arrival of an AL-SDU.

– AL-Abort.request: This primitive is issued to AL2M by an AL2 user to signal that a partially delivered AL-SDU is to be aborted.

C.4.2.2.2 Description of parameters

• AL-SDU: This parameter specifies the unit of information exchanged between AL2M and the AL2 user. Each AL-SDU shall contain an integral number of octets. The length of AL-SDUs may be variable. The maximum length of AL-SDUs that an AL2M receiver can accept shall be signalled via the H.245 control channel. The octets in an AL-SDU are numbered from 1 to n, and in each octet, bits are numbered from 1 to 8. Bit 1 of the octet 1 is transmitted first. An AL2M receiving entity may deliver an empty AL-SDU to the AL2 user to indicate that an AL-SDU is missing.

• Error Indication (EI): This parameter should be used in the AL2M receiver to pass error indications to the AL2 user. This may also be used if the AL2M receiver entity delivers an empty AL-SDU to the AL2 user. The precise procedures for using this parameter, and its numerical coding, are outside the scope of this Recommendation.

C.4.2.3 Format and Coding of AL2M

The format of the AL-PDU is illustrated in Figure C.8. The entire AL-PDU interleaving as described in C.4.2.3.2 shall be used, if this has been required in the H.245 OpenLogicalChannel message.

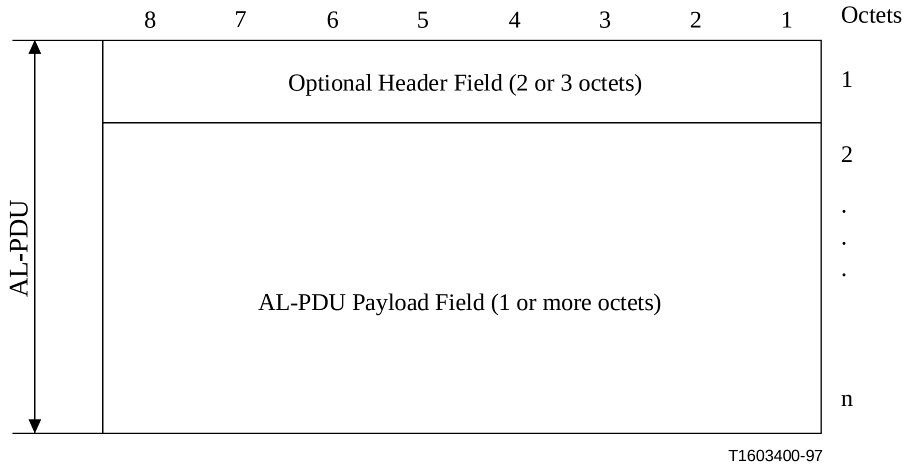


Figure C.8/H.223 – AL-PDU format for AL2M

C.4.2.3.1 Header Field

The optional header consists of the 5-bit or 12-bit Sequence Number (SN) and a corresponding Header Error Control (HEC) field. This HEC uses either an SEBCH(16, 5) or an EGolay Code(24, 12), (see Figures C.9 and C.10).

8	7	6	5	4	3	2	1	Octets
P3	P2	P1	SN5	SN4	SN3	SN2	SN1	1
P11	P10	P9	P8	P7	P6	P5	P4	2

Figure C.9/H.223 – Control field format of the AL-PDU for AL2M with SN = 5 and SEBCH code

8	7	6	5	4	3	2	1	Octets
SN8	SN7	SN6	SN5	SN4	SN3	SN2	SN1	1
P4	P3	P2	P1	SN12	SN11	SN10	SN9	2
P12	P11	P10	P9	P8	P7	P6	P5	3

Figure C.10/H.233 – Control field format of the AL-PDU for AL2M with SN = 12 EGolay code

C.4.2.3.1.1 Sequence Number (SN) field

The optional 5-bit/12-bit SN provides a capability for sequencing AL-PDUs. The sequence number may be used by the AL2M receiving entity to detect missing and misdelivered AL-PDUs.

AL2M receivers conforming to ITU-T H.223 shall be capable of receiving and correctly interpreting AL-PDUs that include the SN field. The use of the SN field shall be determined by the transmitter and shall be signalled to the far end in the OpenLogicalChannel message of H.245.

When the SN field is in use, the AL2M receiver may detect that an AL-PDU is missing or has been misdelivered by the MUX layer. The AL2M receiver should discard any misdelivered AL-PDUs that it detects.

C.4.2.3.1.2 Header Error Control (HEC) field of the AL2M header

The optional AL2M header uses an SEBCH(16, 5) or an EGolay (24, 12) code. The definition of the EGolay code shall be the same as described in C.4.1.5.4, whereby the field RN is replaced by SN11 and X by SN12. The SEBCH code shall be used by the definition of Table I.1. The CEC bits of SEBCH code shall be derived by the following equation:

$$\begin{bmatrix} P1 \\ P2 \\ P3 \\ P4 \\ P5 \\ P6 \\ P7 \\ P8 \\ P9 \\ P10 \\ P11 \end{bmatrix} = \begin{bmatrix} 1\,1\,1\,0\,1\,1\,0\,0\,1\,0\,1 \\ 0\,1\,1\,1\,0\,1\,1\,0\,0\,1\,1 \\ 1\,1\,0\,1\,0\,1\,1\,1\,1\,0\,0 \\ 0\,1\,1\,0\,1\,0\,1\,1\,1\,1\,0 \\ 1\,1\,0\,1\,1\,0\,0\,1\,0\,1\,1 \end{bmatrix}^{T} \bullet \begin{bmatrix} SN1 \\ SN2 \\ SN3 \\ SN4 \\ SN5 \end{bmatrix}$$

C.4.2.3.1.3 AL-PDU Payload Field

The AL-PDU field contains a complete AL-SDU, where the first octet corresponds to the first octet of the AL-SDU. Both the AL-SDU and the AL-PDU are octet aligned.

C.4.2.3.2 Interleaving

If interleaving is required in the H.245 OpenLogicalChannel message, it shall be applied to the entire AL-PDU including the header field. The same interleaver as described in C.4.1.8 shall be used for AL2M.

De-interleaving also has to be applied at receiver side in this case.

C.4.2.4 Procedures for Abort

This primitive is discarded and no action is undertaken.

C.4.2.5 Procedure for Sequence Numbering

The following procedures apply when the SN field is in use.

Once a logical channel using AL2M is successfully opened according to the procedure defined in ITU-T H.245, the first AL-PDU transmitted by the AL2M sending entity shall have the SN field set to 0. For each subsequent transmitted AL-PDU which belongs to that logical channel, the value of the SN field shall be incremented by 1 modulo 32 for a 5-bit SN-field (or modulo 4096 for the 12-bit SN-field).

C.4.2.6 Procedures for Error Control

When the SEBCH/EGolay decoding fails at the AL2M receiver, the associated AL-SDU may be delivered to the AL2 user, together with an appropriate error indication, via the AL-DATA.indication primitive.

When the SN field is in use, the AL2M receiver may detect that an AL-PDU is missing or has been misdelivered by the MUX layer. The AL2M receiver should discard any misdelivered AL-PDUs that it detects. For each missing AL-PDU that it detects, the AL2M receiver may deliver to the AL2 user an empty AL-SDU, together with an appropriate error indication, via the AL-DATA.indication primitive.

C.4.3 AL3M

AL3M is designed primarily for the transfer of video. The format, structure, definitions and procedures are identical to the adaptation layer AL1M, see C.4.1, except that:

- AL3M shall support one framed transfer mode; and
- AL3M shall always operate in splitting mode, while using the ARQ-mode and shall not use splitting mode, while operating in FEC_ONLY mode.

In AL3M it is possible that additional error control can be provided by the higher layer protocol, e.g. by the procedures in Annex N/H.263.

ANNEX D

Optional multiplexing protocol for low bit rate multimedia mobile communication over highly error-prone channels

D.1 General

In this annex, an optional level 3 protocol of the H.223 mobile extensions is specified. In order to maintain compatibility, the basic features of the level 3 protocol described in Annex C shall be included.

D.2 Acronyms

This annex uses the following acronyms:

ARQ Automatic Repeat reQuest

CF Control Header Field

CRC Cyclic Redundancy Check

FEC Forward Error Correction

SRS Shortened Reed-Solomon (code)

D.3 Multiplex (MUX) layer specification

See C.3.

D.4 Adaptation layer

D.4.1 AL1M

D.4.1.1 Framework of AL1M

See C.4.1.1.

D.4.1.2 Primitive exchanged between AL1 and AL1M

See C.4.1.2.

D.4.1.3 Functions of AL1M

AL1M provides the following functions:

- optional error detection and indication;
- optional sequence numbering;
- optional forward error correction;

- optional support of retransmission via ARQI[3];
- optional AL-SDU splitting for framed frames.

D.4.1.4 Format and structure of AL1M

The format of the AL1M can be seen in Figure D.1.

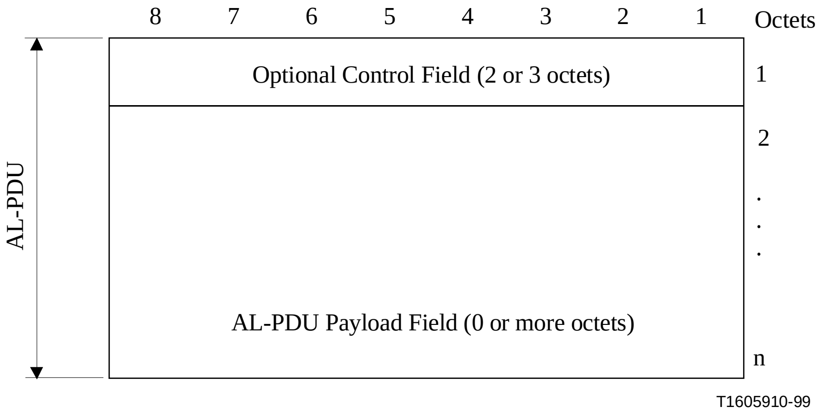


Figure D.1/H.223 – Format of the AL-PDU of the AL1M

The AL-PDU payload shall consist of either an I-PDU or a S-PDU. If a S-PDU is transmitted, the length of the AL-PDU payload is 0, otherwise it is an I-PDU. In the following descriptions, the AL-PDU payload is assumed to be an I-PDU unless some other indication is given. The maximum length of AL-PDUs that an AL1M receiver can accept shall be signalled via the H.245 capability exchange.

In contrast to AL1 of ITU-T H.223, the AL-SDU is not always directly mapped to the AL-PDU payload, see Figure D.2. The application layer (AL1 user) transfers its data through AL-SDUs to the adaptation layer. The adaptation layer forms its own AL-SDUs* from the AL-SDUs. The length of the AL-PDU can be derived from the procedure given in D.4.1.7.1. The AL-PDU is formed by the AL-PDU payload and the optional Control Field (CF).

[3] Note that ARQII is not supported.

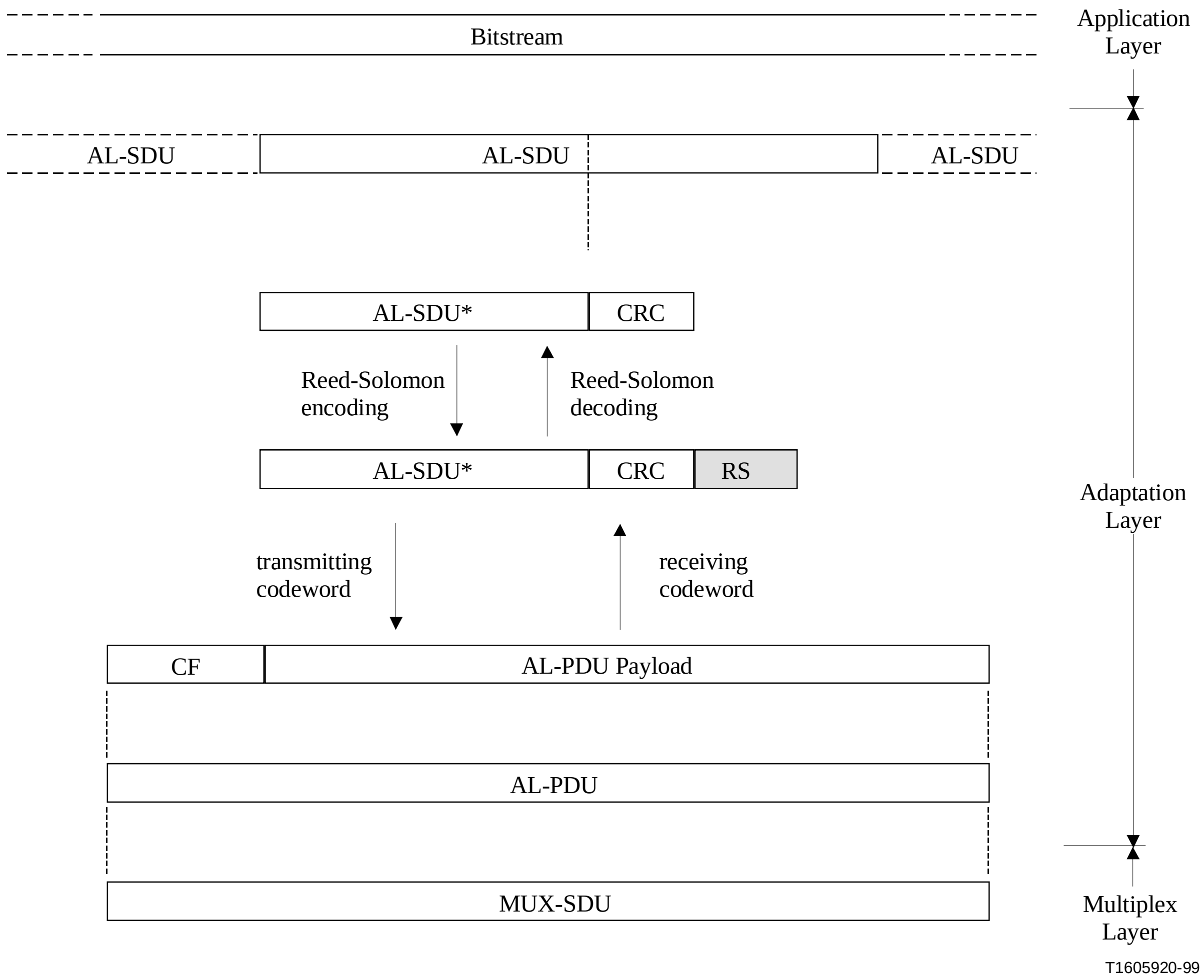


Figure D.2/H.223 – AL1M structure

The error protocol allows the AL1M to operate the following two modes:

FEC_ONLY In this mode an AL-SDU* with a CRC is Reed-Solomon encoded with a code rate $r \leq 1.0$. The length of the AL-SDU* shall be shorter than $255 - 2e_{target} - l_{CRC}/8$. The resulting AL-PDU consists only of an AL-PDU payload field. Splitting mode is not supported. In this mode, no retransmission is possible.

ARQ If the mode is set to ARQI (only ARQI supported), it is possible to request retransmissions.

When ARQI is used, each (re)transmission shall contain the same encoded data. Therefore, the AL-PDU of each retransmission of the same SN shall contain the identical number of octets.

D.4.1.5 Control Field (CF)

See C.4.1.5.

D.4.1.6 Procedures for splitting an AL-SDU (splitting mode)

Only in framed transfer mode, the adaptation layer may split the AL-SDU into one or several AL-SDU*s if the use of this splitting procedure is signalled by the OpenLogicalChannel message. This procedure is mandatory for the receiver. In the event that AL-SDU is longer than $255 - 2e_{target} - l_{CRC}/8$ octets, transmitter shall apply this splitting procedure. In the event that AL-SDU is shorter than $255 - 2e_{target} - l_{CRC}/8$ octets, transmitter may apply this splitting procedure.

Each AL-SDU* is transmitted as described in D.4.1.7. To identify the end of an AL-SDU, the last AL-SDU* of the AL-SDU shall be marked by setting the RN field to logical "1", otherwise the RN field shall be set to "0".

D.4.1.7 Procedures for encoding and decoding the AL-PDU payload

See C.4.1.7.

D.4.1.7.1 Evaluation of the AL-PDU (I-PDU) length

The following parameters are given:

- l_v length of AL-PDU in bits;
- t length of AL-SDU* in bits;
- e_{target} correction ability of the SRS code in octets;
- l_h length of the control header (CF) field in bits;
- l_{CRC} length of the Cyclic Redundancy Check (CRC) field in bits.

The length l_v of the AL-PDU can be evaluated by the following equation:

$$l_v = l_h + t + l_{CRC} + 16e_{target} \tag{D-1}$$

The parameters l_v, t and l_{CRC} shall be byte aligned. Equation (D-1) shall be used by the AL1M transmitter. At the AL1M receiver the length of the AL-SDU* t shall be evaluated by the following equation:

$$t = l_v - l_h - l_{CRC} - 16e_{target} \tag{D-2}$$

Both equations shall be calculated in octets, as illustrated by the following example:

Example

The AL1M wants to transmit an AL-SDU* of t = 376 bits (47 octets), e_{target} = 2, l_h = 24 bits (3 octets), l_{CRC} = 16 bits (2 octets). Using the equation (D-1), the length of the AL-PDU is l_v = 56 octets. The instant rate r_{result} can be evaluated by:

$$r_{result} = \frac{t + l_{CRC}}{l_v - l_h} \tag{D-3}$$

In this example $r_{result} = \frac{49}{53} \approx 0.9245$.

D.4.1.7.2 Cyclic Redundancy Check (CRC)

The CRC provides error detection capability across the entire AL-SDU*, however no CRC may be used. The CRC is appended to the AL-SDU* before the error correction coding procedure is done. The CRC is used by the AL1M receiver to verify whether the decoding attempt of the error correction algorithm is error-free. CRC lengths of 8, 16 and 32 bits are supported. The length of the CRC field shall be specified during the H.245 OpenLogicalChannel procedure.

Description of the CRC polynomials:

a) 8-bit CRC: see 7.3.3.2.3;

b) 16-bit CRC: see 7.4.3.2.3;

c) 32-bit CRC: see 8.1.1.6.2/V.42.

D.4.1.7.3 Shortened Reed-Solomon encoder

The channel encoder is based on a Shortened Reed-Solomon (SRS) encoder with correction ability e_{target}, where e_{target} can be selected as an arbitrary integer value satisfying $0 \le 2e_{target} \le 255 - (t + l_{CRC})/8$, where t and l_{CRC} denote length of AL-SDU* and length of CRC, respectively. At the AL1M sending unit, the AL-PDU payload is generated by Reed-Solomon encoding of the concatenated field of the AL-SDU* and CRC field. The Reed-Solomon encoding of the CRC field starts with the highest order term of the polynomial representing the CRC field. At the AL1M receiving entity, the concatenation of AL-SDU* and CRC field may be reconstructed by Reed-Solomon decoding. As this code is systematic the receiver may also directly extract the CRC protected AL-SDU* from the received bitstream without Reed-Solomon decoding. The SRS code defined in the Galois field GF(2^8) is generated from a generator polynomial $\mathrm{g(x)} = (\mathrm{x} - \alpha)(\mathrm{x} - \alpha^2)\ldots(\mathrm{x} - \alpha^{2\mathrm{e}_{\mathrm{target}}})$, where $\alpha^{\mathrm{i}}(0 \le \mathrm{i} \le 254)$ denotes a root of the primitive polynomial $m(x) = x^8 + x^4 + x^3 + x^2 + 1$. Table II.1 shows binary 8-tuple representations for α^i. A shift register realization is shown in Figure D.3.

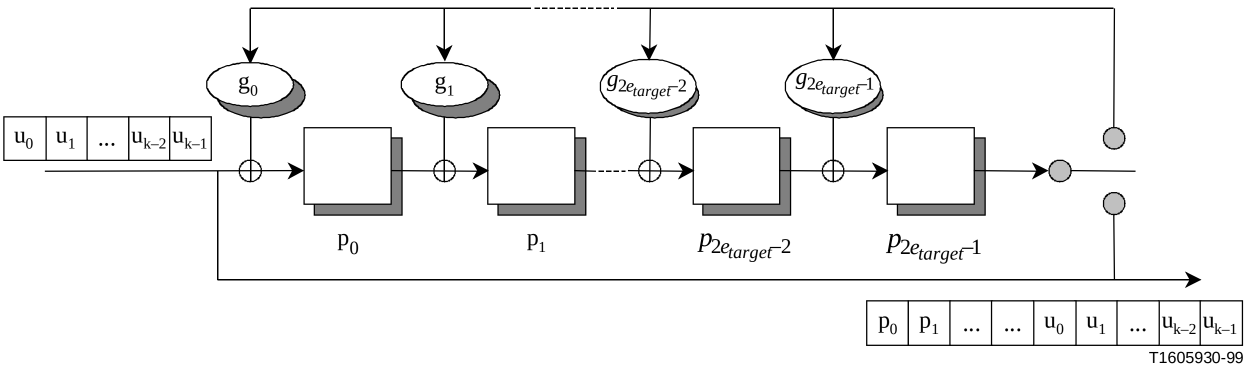


Figure D.3/H.223 – Shift register realization for Reed-Solomon encoder

Each element of the message sequence $\boldsymbol{u} = (u_{k-1}, u_{k-2}, k, u_1, u_0)$ corresponds to that of AL-SDU* and CRC in octets. The parity check polynomial $p(x)$ is calculated as:

$$\begin{aligned} p(x) &= x^{2e_{target}} \cdot u(x) \bmod g(x) \\ &= p_{2e_{target}-1} x^{2e_{target}-1} + p_{2e_{target}-2} x^{2e_{target}-2} + \Lambda + p_1 x + p_0 \end{aligned} \qquad \text{(D-4)}$$

where $u(x)$ denotes the message polynomial defined as:

$$u(x) = u_{k-1} x^{k-1} + u_{k-2} x^{k-2} + \Lambda + u_1 x + u_0 \qquad \text{(D-5)}$$

From (D-4) and (D-5), the code polynomial is given by:

$$\begin{aligned} c(x) = {} & u_{k-1} x^{2e_{target}+k-1} + u_{k-2} x^{2e_{target}+k-2} + \Lambda + u_1 x^{2e_{target}+1} + u_0 x^{2e_{target}} \\ & + p_{2e_{target}-1} x^{2e_{target}-1} + p_{2e_{target}-2} x^{2e_{target}-2} + \Lambda\ p_1 x + p_0 \end{aligned} \qquad \text{(D-6)}$$

Example

- $e_{target} = 2$
- $u = (u_2, u_1, u_0) = (\alpha^4, \alpha^7, \alpha^{231})$
- $l_{CRC} = 8$

In this example, u_2, and u_1 are assumed to be AL-SDU* and u_0 be CRC. According to the procedure of 7.3.3.2.3, CRC polynomial $b(x)$ is given by:

$$b(x) = x^7 + x^5 + x^3 + x^2 + x + 1 \qquad \text{(D-7)}$$

Then, $u_0 = \alpha^{231}$ is obtained.

The generator polynomial $g(x)$ is given by:

$$\begin{aligned} g(x) &= (x-\alpha)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4) \\ &= x^4 + \alpha^{76}x^3 + \alpha^{251}x^2 + \alpha^{81}x + \alpha^{10} \end{aligned} \qquad \text{(D-8)}$$

Each element of the message sequence $u = (\alpha^4, \alpha^7, \alpha^{231})$ corresponds to that of AL-SDU* and CRC in octets. The parity check polynomial $p(x)$ is then calculated as:

$$\begin{aligned} p(x) &= x^4\left(\alpha^4x^2 + \alpha^7x + \alpha^{231}\right) \bmod g(x) \\ &= \alpha^{34}x^3 + \alpha^{12}x^2 + \alpha^{189}x + \alpha^{188} \end{aligned} \qquad \text{(D-9)}$$

From (D-8) and (D-9), the code polynomial is given by:

$$c(x) = \alpha^4x^6 + \alpha^7x^5 + \alpha^{231}x^4 + \alpha^{34}x^3 + \alpha^{12}x^2 + \alpha^{189}x + \alpha^{188} \qquad \text{(D-10)}$$

Therefore, the code sequence $c = (\alpha^4, \alpha^7, \alpha^{231}, \alpha^{34}, \alpha^{12}, \alpha^{189}, \alpha^{188})$ is obtained.

Figure D.4 shows a shift register realization of this example.

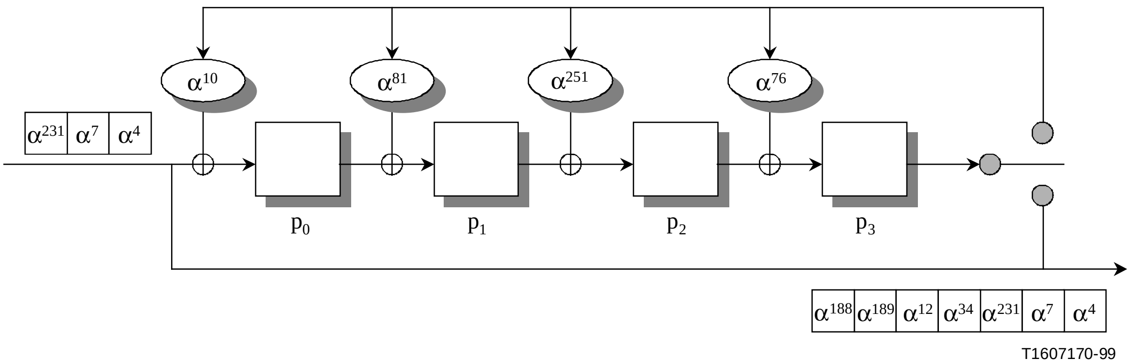


Figure D.4/H.223 – Example of Reed-Solomon encoder (e_{target} = 2)

D.4.1.8 Encoding procedure: AL-SDU* (I-PDU) to AL-PDU

The following steps are necessary to obtain an AL-PDU from an AL-SDU*.

1) The CRC of the length required in the H.245 OpenLogicalChannel message field shall be added to the AL-SDU*.
2) Generate the encoded data by passing the AL-SDU* plus CRC through the Reed-Solomon encoder.

3) For the first transmission read the highest order term of the code polynomial (e.g. u_{k-1} in Figure D.3). The first octet of the output (e.g. u_{k-1} in Figure D.3) is the first octet of the AL-PDU payload field.

4) If required (as indicated in the H.245 OpenLogicalChannel message), the Control Field (CF) shall be added at the beginning of the AL-PDU.

These steps are valid for the modes FEC_ONLY and ARQI.

Figure D.5 illustrates the encoding procedures of the AL1M at the transmitter side.

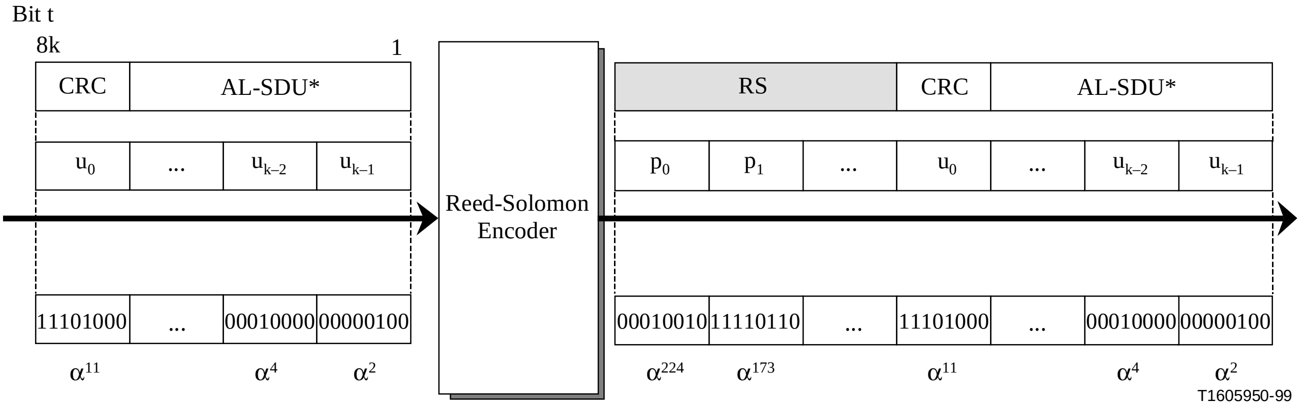


Figure D.5/H.223 – Encoding procedure of the AL1M at the transmitter side

D.4.1.9 Decoding of the AL-PDU payload (I-PDU)

The receiver may check the received systematic symbols before decoding the Reed-Solomon code. If CRC check fails, any kind of Reed-Solomon decoding may be used.

After Reed-Solomon decoding, the CRC may be used to check the correctness of the decoding attempt. If the CRC fails, another retransmission may be requested, or the wrong data may be given to the AL1M user with an appropriate Error Indication (EI) message. If error correction fails, the receiver may use the decoded information symbols or the systematic symbols before Reed-Solomon decoding as received in the AL-SDU*. Again passing the wrong data to the AL1M user along with an EI message.

If ARQI retransmission procedure is used, each retransmission gives the same data as the previous one. After each decoding attempt, the decoding result may be checked by the CRC.

D.4.1.10 Procedures for Abort

See C.4.1.11.

D.4.1.11 Procedures for error control

See C.4.1.12.

D.4.1.12 Retransmission procedures (ARQI)

See C.4.1.13.

D.4.2 AL2M

See C.4.2.

D.4.3 AL3M

See C.4.3. AL3M in Annex D shall use SRS code in stead of RCPC code.

APPENDIX I

Generator matrixes of the Systematic Extended BCH

This appendix describes Systematic Extended Bose-Chaudhuri-Hocquenghem (SEBCH) codes and includes the generator matrixes, which are used in Annex C.

I.1 BCH codes

BCH codes are linear cyclic block codes, hence they can be described using a generator polynomial. However, the easiest way to describe short block codes is using a generator matrix which describes all characteristics of the code. With a generator matrix $\underline{\boldsymbol{G}}$ and a information sequence $\underline{\boldsymbol{i}}$ of length $\boldsymbol{k}$ the code vector $\underline{\boldsymbol{c}}$ of length $\boldsymbol{n}$ can be obtained by:

$$\underline{\mathbf{c}} = \underline{\mathbf{i}} \cdot \underline{\mathbf{G}} = [\, \underline{\mathbf{i}}^{\mathbf{T}} \mid \underline{\mathbf{c}_0}^{\mathbf{T}} \,]^{\mathbf{T}}$$

with $\underline{\boldsymbol{G}} = [\ \underline{\mathbf{1}} \mid \underline{\boldsymbol{A}}\ ]$ a $(\boldsymbol{k} \times \boldsymbol{n})$ matrix containing a $(\boldsymbol{k} \times \boldsymbol{k})$ identity matrix in the first $\boldsymbol{k}$ columns/rows to obtain a systematic code. For a primitive BCH code the length of the code $\boldsymbol{n}$ is always $\boldsymbol{n} = \mathbf{2}^{\mathbf{h}} - \mathbf{1}$. For $\boldsymbol{k}$ there are some constraints, not all values are possible.

The third parameter describing a block code besides code length $\boldsymbol{n}$ and information length $\boldsymbol{k}$ is the minimum distance between two code words $\boldsymbol{d}$. If a code has minimal distance $\boldsymbol{d}$, it can correct at most $\lfloor (\boldsymbol{d} - \mathbf{1}) / \mathbf{2} \rfloor$ errors or detect $(\boldsymbol{d} - \mathbf{1})$ errors.

I.2 Systematic Extended BCH codes

As all linear cyclic block codes can be made systematic, there always exits a systematic BCH code.

As we evaluated earlier, primitive BCH codes always have the length $\boldsymbol{n} = \mathbf{2}^{\mathbf{h}} - \mathbf{1}$. To make these codes octet aligned, extension has to be applied. The extension of a BCH($\boldsymbol{n}, \boldsymbol{k}, \boldsymbol{d}$) has the length $\boldsymbol{n} + \mathbf{1}$. One digit is appended, so that each code word has even weight. The extended BCH code then always has minimal distance $\boldsymbol{d} + \mathbf{1}$. Hence we derived from BCH($\boldsymbol{n}, \boldsymbol{k}, \boldsymbol{d}$) a code EXBCH($\boldsymbol{n} + \mathbf{1}, \boldsymbol{k}, \boldsymbol{d} + \mathbf{1}$). Extended codes are still linear, but no more cyclic. Hence the description using generator polynomials is impossible.

The generator matrix of the extended code from $\underline{\boldsymbol{G}}$ of the mother code can be derived by adding one column which contains the parity check bit of each row. The examples of the generator matrices are given in Tables I.1 and I.2.

I.3 Decoder overview

For decoding BCH codes, usually Berleykamp-Massey algorithm is used. This is an efficient method to determine error locations in the received vector. There are also some approaches to use reliability information for decoding block codes. However, these algorithms yield in high complexity.

One main feature of BCH codes is the possibility to use these codes for error correction and detection at the same time. For example a code with $\boldsymbol{d} = \mathbf{5}$ could correct up to one error and detect up to three errors in parallel. With the usage of BCH codes only, the decoder has the flexibility to decide how many errors to correct and use the rest of redundancy for error detection. Berleykamp-Massey algorithm can also be used for this.

Example

In this example we use the SEBCH(16, 5, 8). The information vector $\underline{\boldsymbol{i}}$ is given as:

$$\underline{\boldsymbol{i}} = [1\ 0\ 0\ 1\ 1]$$

By using the generator matrix $\underline{\boldsymbol{G}}$, the code word $\underline{\boldsymbol{c}}$ can be evaluated by:

$$\underline{\boldsymbol{c}} = \underline{\boldsymbol{i}} \cdot \underline{\boldsymbol{G}} = [1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0]$$

For transmission these bits are filled into octet-aligned fields. The LSB-bit of the vector $\underline{\boldsymbol{c}}$ is at its left side, the MSB at its right. The LSB of $\underline{\boldsymbol{c}}$ is filled to the lowest numbered bit of the last octet (octet 2) and the MSB of $\underline{\boldsymbol{c}}$ to the highest-numbered bit of the first octet (octet 1) (see Figure I.1).

8	7	6	5	4	3	2	1	Octets
0	0	0	0	1	1	1	1	1
0	1	0	1	1	0	0	1	2

Figure I.1/H.223 – Field mapping convention of SEBCH codes

I.4 Generator matrices for Systematic Extended BCH codes

In this clause we provide some tables to calculate a code sequence $\underline{\boldsymbol{c}}$ of length $\boldsymbol{n}$ from a given input sequence $\underline{\boldsymbol{i}}$ of length $\boldsymbol{k}$ using the generator matrix $\underline{\boldsymbol{G}}$ with the equation: $\underline{\boldsymbol{c}} = \underline{\boldsymbol{i}} \cdot \underline{\boldsymbol{G}}$. SEBCH(16, 5, 8) is derived from BCH(15, 5, 7) with generator polynomial $g(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x^1 + 1$, SEBCH(16, 7, 6) is derived from BCH(15, 7, 5) with generator polynomial $g(x) = x^8 + x^7 + x^6 + x^4 + 1$.

Table I.1/H.223 – Generator matrix for Systematic Extended BCH(16, 5, 8) code

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	1	0	0	0	0	1	1	1	0	1	1	0	0	1	0	1
1	0	1	0	0	0	0	1	1	1	0	1	1	0	0	1	1
2	0	0	1	0	0	1	1	0	1	0	1	1	1	1	0	0
3	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1	0
4	0	0	0	0	1	1	1	0	1	1	0	0	1	0	1	1

Table I.2/H.223 – Generator matrix for Systematic Extended BCH(16, 7, 6) code

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	1	0	0	0	0	0	0	1	0	0	0	1	0	1	1	1
1	0	1	0	0	0	0	0	1	1	0	0	1	1	1	0	0
2	0	0	1	0	0	0	0	0	1	1	0	0	1	1	1	0
3	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1
4	0	0	0	0	1	0	0	0	1	0	1	1	1	0	0	1
5	0	0	0	0	0	1	0	0	0	1	0	1	1	1	0	1
6	0	0	0	0	0	0	1	0	0	0	1	0	1	1	1	1

APPENDIX II

Binary representation for α^i

This appendix expresses the binary representation for α^i over GF(2^8) used in Annex D. In a binary represented α^i ($u^{(8)}$, $u^{(7)}$, $u^{(6)}$, $u^{(5)}$, $u^{(4)}$, $u^{(3)}$, $u^{(2)}$, $u^{(1)}$), $u^{(1)}$ is defined as LSB and $u^{(8)}$ as MSB. See Table II.1.

Table II.1/H.223 – Binary representation for α^i ($0 \le i \le 254$) over GF(2^8)

α^i	Binary rep.	α^i	Binary rep.	α^i	Binary rep.	α^i	Binary rep.
0	00000000	α^{63}	10100001	α^{127}	11001100	α^{191}	01000001
α^{0}	00000001	α^{64}	01011111	α^{128}	10000101	α^{192}	10000010
α^{1}	00000010	α^{65}	10111110	α^{129}	00010111	α^{193}	00011001
α^{2}	00000100	α^{66}	01100001	α^{130}	00101110	α^{194}	00110010
α^{3}	00001000	α^{67}	11000010	α^{131}	01011100	α^{195}	01100100
α^{4}	00010000	α^{68}	10011001	α^{132}	10111000	α^{196}	11001000
α^{5}	00100000	α^{69}	00101111	α^{133}	01101101	α^{197}	10001101
α^{6}	01000000	α^{70}	01011110	α^{134}	11011010	α^{198}	00000111
α^{7}	10000000	α^{71}	10111100	α^{135}	10101001	α^{199}	00001110
α^{8}	00011101	α^{72}	01100101	α^{136}	01001111	α^{200}	00011100
α^{9}	00111010	α^{73}	11001010	α^{137}	10011110	α^{201}	00111000
α^{10}	01110100	α^{74}	10001001	α^{138}	00100001	α^{202}	01110000
α^{11}	11101000	α^{75}	00001111	α^{139}	01000010	α^{203}	11100000
α^{12}	11001101	α^{76}	00011110	α^{140}	10000100	α^{204}	11011101
α^{13}	10000111	α^{77}	00111100	α^{141}	00010101	α^{205}	10100111
α^{14}	00010011	α^{78}	01111000	α^{142}	00101010	α^{206}	01010011
α^{15}	00100110	α^{79}	11110000	α^{143}	01010100	α^{207}	10100110
α^{16}	01001100	α^{80}	11111101	α^{144}	10101000	α^{208}	01010001
α^{17}	10011000	α^{81}	11100111	α^{145}	01001101	α^{209}	10100010
α^{18}	00101101	α^{82}	11010011	α^{146}	10011010	α^{210}	01011001
α^{19}	01011010	α^{83}	10111011	α^{147}	00101001	α^{211}	10110010
α^{20}	10110100	α^{84}	01101011	α^{148}	01010010	α^{212}	01111001
α^{21}	01110101	α^{85}	11010110	α^{149}	10100100	α^{213}	11110010
α^{22}	11101010	α^{86}	10110001	α^{150}	01010101	α^{214}	11111001
α^{23}	11001001	α^{87}	01111111	α^{151}	10101010	α^{215}	11101111
α^{24}	10001111	α^{88}	11111110	α^{152}	01001001	α^{216}	11000011
α^{25}	00000011	α^{89}	11100001	α^{153}	10010010	α^{217}	10011011
α^{26}	00000110	α^{90}	11011111	α^{154}	00111001	α^{218}	00101011

Table II.1/H.223 – Binary representation for α^i ($0 \le i \le 254$) over GF(2^8)

α^i	Binary rep.	α^i	Binary rep.	α^i	Binary rep.	α^i	Binary rep.
α^{27}	00001100	α^{91}	10100011	α^{155}	01110010	α^{219}	01010110
α^{28}	00011000	α^{92}	01011011	α^{156}	11100100	α^{220}	10101100
α^{29}	00110000	α^{93}	10110110	α^{157}	11010101	α^{221}	01000101
α^{30}	01100000	α^{94}	01110001	α^{158}	10110111	α^{222}	10001010
α^{31}	11000000	α^{95}	11100010	α^{159}	01110011	α^{223}	00001001
α^{32}	10011101	α^{96}	11011001	α^{160}	11100110	α^{224}	00010010
α^{33}	00100111	α^{97}	10101111	α^{161}	11010001	α^{225}	00100100
α^{34}	01001110	α^{98}	01000011	α^{162}	10111111	α^{226}	01001000
α^{35}	10011100	α^{99}	10000110	α^{163}	01100011	α^{227}	10010000
α^{36}	00100101	α^{100}	00010001	α^{164}	11000110	α^{228}	00111101
α^{37}	01001010	α^{101}	00100010	α^{165}	10010001	α^{229}	01111010
α^{38}	10010100	α^{102}	01000100	α^{166}	00111111	α^{230}	11110100
α^{39}	00110101	α^{103}	10001000	α^{167}	01111110	α^{231}	11110101
α^{40}	01101010	α^{104}	00001101	α^{168}	11111100	α^{232}	11110111
α^{41}	11010100	α^{105}	00011010	α^{169}	11100101	α^{233}	11110011
α^{42}	10110101	α^{106}	00110100	α^{170}	11010111	α^{234}	11111011
α^{43}	01110111	α^{107}	01101000	α^{171}	10110011	α^{235}	11101011
α^{44}	11101110	α^{108}	11010000	α^{172}	01111011	α^{236}	11001011
α^{45}	11000001	α^{109}	10111101	a^{173}	11110110	α^{237}	10001011
α^{46}	10011111	α^{110}	01100111	α^{174}	11110001	α^{238}	00001011
α^{47}	00100011	α^{111}	11001110	α^{175}	11111111	α^{239}	00010110
α^{48}	01000110	α^{112}	10000001	α^{176}	11100011	α^{240}	00101100
α^{49}	10001100	α^{113}	00011111	α^{177}	11011011	α^{241}	01011000
α^{50}	00000101	α^{114}	00111110	α^{178}	10101011	α^{242}	10110000
α^{51}	00001010	α^{115}	01111100	α^{179}	01001011	α^{243}	01111101
α^{52}	00010100	α^{116}	11111000	α^{180}	10010110	α^{244}	11111010
α^{53}	00101000	α^{117}	11101101	α^{181}	00110001	α^{245}	11101001
α^{54}	01010000	α^{118}	11000111	α^{182}	01100010	α^{246}	11001111
α^{55}	10100000	α^{119}	10010011	α^{183}	11000100	α^{247}	10000011
α^{56}	01011101	α^{120}	00111011	α^{184}	10010101	α^{248}	00011011
α^{57}	10111010	α^{121}	01110110	α^{185}	00110111	α^{249}	00110110
α^{58}	01101001	α^{122}	11101100	α^{186}	01101110	α^{250}	01101100
α^{59}	11010010	α^{123}	11000101	α^{187}	11011100	α^{251}	11011000

Table II.1/H.223 – Binary representation for α^i ($0 \leq i \leq 254$) over GF(2^8)

α^i	Binary rep.	α^i	Binary rep.	α^i	Binary rep.	α^i	Binary rep.
α^{60}	10111001	α^{124}	10010111	α^{188}	10100101	α^{252}	10101101
α^{61}	01101111	α^{125}	00110011	α^{189}	01010111	α^{253}	01000111
α^{62}	11011110	α^{126}	01100110	α^{190}	10101110	α^{254}	10001110

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2002