# TECHNICAL SPECIFICATION

# IEC TS 61334-5-4

First edition
2001-06

**Distribution automation using distribution line carrier systems –**

**Part 5-4:
Lower layer profiles –
Multi-carrier modulation (MCM) profile**



Reference number
IEC/TS 61334-5-4:2001(E)

## Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

## Consolidated editions

The IEC is now publishing consolidated versions of its publications. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

## Further information on IEC publications

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology. Information relating to this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is also available from the following:

- **IEC Web Site (www.iec.ch)**

- **Catalogue of IEC publications**

  The on-line catalogue on the IEC web site (www.iec.ch/catlg-e.htm) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.

- **IEC Just Published**

  This summary of recently issued publications (www.iec.ch/JP.htm) is also available by email. Please contact the Customer Service Centre (see below) for further information.

- **Customer Service Centre**

  If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

  Email: custserv@iec.ch
  Tel: +41 22 919 02 11
  Fax: +41 22 919 03 00

# TECHNICAL SPECIFICATION **IEC TS 61334-5-4**

First edition
2001-06

## Distribution automation using distribution line carrier systems –

## Part 5-4: Lower layer profiles – Multi-carrier modulation (MCM) profile





International Electrotechnical Commission 3, rue de Varembé Geneva, Switzerland
Telefax: +41 22 919 0300 e-mail: inmail@iec.ch IEC web site http://www.iec.ch

Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

PRICE CODE W

*For price, see current catalogue*

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

# DISTRIBUTION AUTOMATION USING DISTRIBUTION LINE CARRIER SYSTEMS –

## Part 5-4: Lower layer profiles – Multi-carrier modulation (MCM) profile

## FOREWORD

1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, the IEC publishes International Standards. Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. The IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of the IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested National Committees.
3) The documents produced have the form of recommendations for international use and are published in the form of standards, technical specifications, technical reports or guides and they are accepted by the National Committees in that sense.
4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.
5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.
6) Attention is drawn to the possibility that some of the elements of this technical specification may be the subject of patent rights. The IEC shall not be held responsible for identifying any or all such patent rights.

The main task of IEC technical committees is to prepare International Standards. In exceptional circumstances, a technical committee may propose the publication of a technical specification when

- the required support cannot be obtained for the publication of an International Standard, despite repeated efforts, or
- The subject is still under technical development or where, for any other reason, there is the future but no immediate possibility of an agreement on an International Standard.

IEC 61334-5-4, which is a technical specification, has been prepared by IEC technical committee 57: Power system control and associated communications.

The text of this technical specification is based on the following documents:

| Enquiry draft | Report on voting |
|---|---|
| 57/479/CDV | 57/517/RVC |

Full information on the voting for the approval of this technical specification can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 3.

A bilingual version of this publication may be issued at a later date.

The committee has decided that the contents of this publication will remain unchanged until 2004. At this date, the publication will be

- transformed into an International Standard;
- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

# DISTRIBUTION AUTOMATION USING DISTRIBUTION LINE CARRIER SYSTEMS –

# Part 5-4: Lower layer profiles – Multi-carrier modulation (MCM) profile

## 1 Scope and object

This technical specification describes the requirements of the multicarrier modulation (MCM) approach which incorporates the services provided by the physical layer entity and the MAC sublayer with the purpose of building up a set of standards for effective communication on MV and LV network for distribution line carrier (DLC) systems, in the context of IEC 61334-1-1.

Different technical approaches in developing communication systems for DLC communication are in progress. As a consequence, at present, different lower layer profiles are feasible with acceptable results in terms of performance and cost-effectiveness. In many cases, the differences amongst solutions are minor and it is possible to find a common root.

## 2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of IEC 61334. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of IEC 61334 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of IEC and ISO maintain registers of currently valid International Standards.

IEC 61334-1-1, *Distribution automation using distribution line carrier systems – Part 1: General considerations – Section 1: Distribution automation system architecture*

IEC 61334-3-1, *Distribution automation using distribution line carrier systems – Part 3-1: Mains signalling requirements – Frequency bands and output levels*

IEC 61334-4-1, *Distribution automation using distribution line carrier systems – Part 4: Data communication protocols – Section 1: Reference model of the communication system*

## 3 Definitions and abbreviations

### 3.1 Definitions

For the purpose of this part of IEC 61334, the following definitions apply.

**3.1.1**
**control direction**
communication direction from the central system to a field device

**3.1.2**
**domain**
logical section of a DLC communication network

**3.1.3**
**hops**
number of *routing repetitions* required for communication between the master and a specific station

**3.1.4**
**initiator**
a station that controls medium access for one *domain*. The *master station* may delegate its 'initiatorship' for a limited time to one of the *slave stations* registered in its domain

NOTE Being an initiator is a dynamic property of a station.

**3.1.5**
**initiator PDU**
a PDU that is sent from an *initiator* to a *non-initiator*, possibly using *routing repeaters* for multi-hop communication

**3.1.6**
**master station**
*station* that works as communication master for a *domain*

NOTE Being a master station is a static property of a station.

**3.1.7**
**monitoring direction**
communication direction from a field device to the central system

**3.1.8**
**non-initiator**
a *station* that is not in the initiator role

NOTE Being a non-initiator is a dynamic property of a station.

**3.1.9**
**non-initiator PDU**
a PDU that is sent from a *non-initiator* to an *initiator*, possibly using *routing repeaters* for multi-hop transmission

NOTE Non-initiator PDUs are only sent in reaction to *initiator PDUs*.

**3.1.10**
**routing repetition**
re-sending a PDU with a modified address field because the destination station cannot communicate directly with the source station. The routing repetition procedure does not involve a network layer but is located in the MAC sublayer instead. A synonymous for routing repetitions is forwarding in the mobile communications context

**3.1.11**
**slave station**
*station* that works as a communication slave within a *domain*. It normally operates as *non-initiator*, but may be switched to operate as *initiator*

NOTE Being a slave station is a static property of a station.

### 3.2 Abbreviations

| | |
|---|---|
| DLC | Distribution line carrier |
| DMT | Discret multitone |
| HV | High voltage |
| LLC | Logical link control |

| | |
|---|---|
| LMI | Layer management interface |
| LV | Low voltage |
| M_SDU | MAC layer service data unit |
| MCM | Multicarrier modulation |
| MIB | Management information base |
| MV | Medium voltage |
| OFDM | Orthogonal frequency division multiplex |
| P_SDU | Physical layer service data unit |
| PDU | Protocol data unit |
| SDU | Service data unit |
| SMAP | System management application process |



## 4 Lower layer profile structure

The MCM lower layer profile exhibits the structure shown in the following figure. This technical specification describes the function of the physical layer and the MAC sublayer.

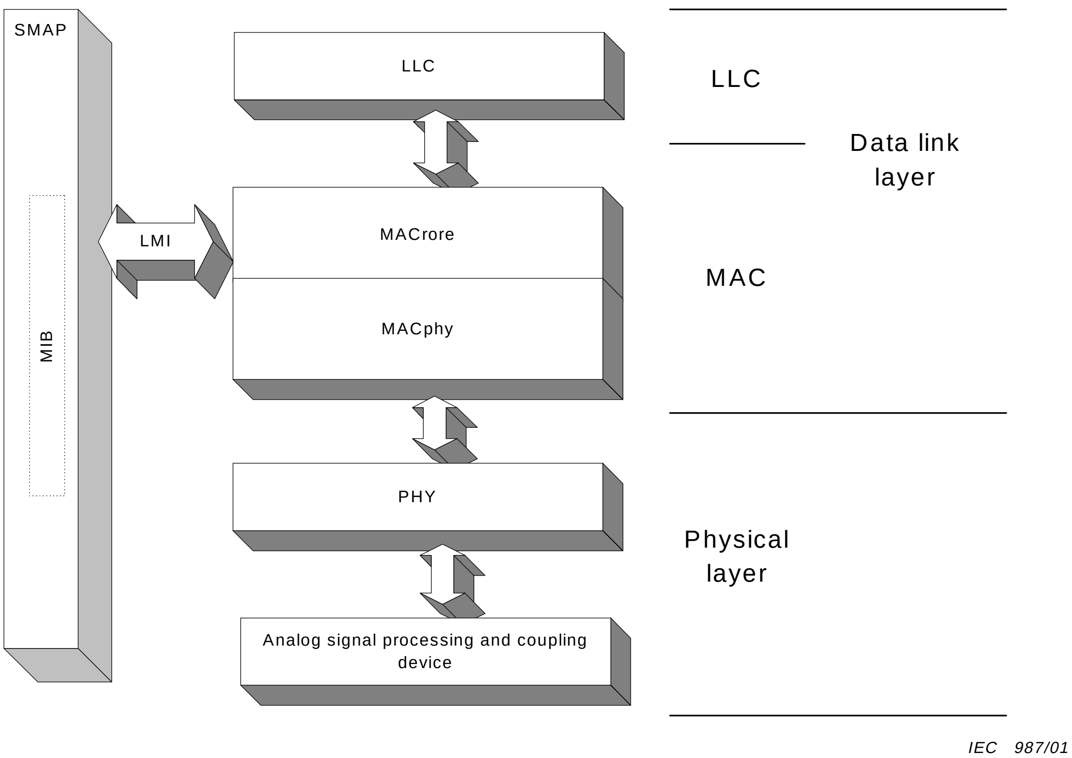



**Figure 1 – Layered architecture of the DLC-M protocol stack**

### 4.1 Physical layer

The physical layer provides services to the MAC sublayer to transfer a MAC protocol data unit to a remote MAC sublayer entity. It is independent of the physical characteristics and the implementation of the mains attachment unit.

### 4.2 MAC sublayer

The MAC sublayer provides services to the LLC sublayer and uses services of the physical layer to transmit LLC PDUs to a remote station. The main functions of the MAC sublayer are error detection and control of medium access.

Furthermore, it provides means for repeater usage which is transparent to the higher protocol layers.

For better understanding, the MAC sublayer is further subdivided into two functional units denoted as MACphy and MACrore. MACphy denotes the part of the MAC sublayer responsible for interfacing to the physical layer, whereas MACrore denotes the part of the MAC sublayer that interfaces the LLC sublayer and is responsible for addressing and routing repetitions.



## 5 Physical layer specification

### 5.1 Modulation

#### 5.1.1 Purpose

Multicarrier modulation (MCM), also known as orthogonal frequency division multiplex (OFDM) or discrete multitone (DMT) is a modulation technique which combines an excellent bandwidth efficiency (high data rates) with the possibility of a very flexible bandwidth allocation. In combination with error correction coding, MCM is very robust in presence of narrowband jammers, impulsive noise, and frequency selective attenuation, as typically seen on power lines.

#### 5.1.2 The multicarrier modulation (MCM) principle

In multicarrier modulation, the channel bandwidth is divided into a number of sub-channels. In each sub-channel, a carrier is modulated at a much lower data-rate. A multicarrier modulation scheme can be viewed as consisting of N independently modulated carriers with different carrier frequencies. If the carrier frequencies are selected appropriately, the various carriers are orthogonal, so that they do not interfere with each other. A sample representation of a multicarrier modulated signal in the frequency is shown in figure 2.

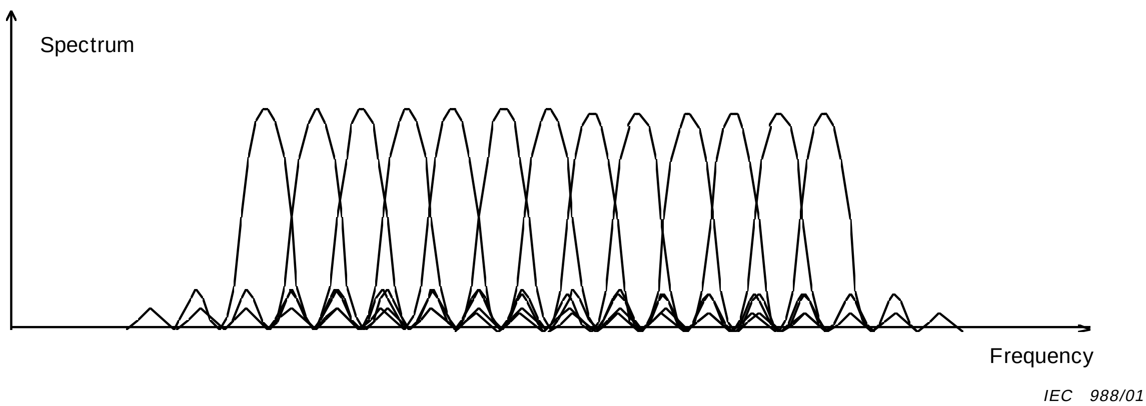



**Figure 2 – Sample frequency representation of multicarrier modulation**

There are several advantages of the multicarrier modulation scheme as compared to traditional single carrier or spread spectrum systems:

– MCM achieves a much higher bandwidth efficiency than spread spectrum systems. If the bandwidth of each carrier is sufficiently small, a data-rate close to the theoretical Shannon limit can be achieved;

- MCM allows an extremely flexible allocation and use of a given channel bandwidth. As an example, the lower and the upper limit of the used frequency band can be easily configured. In addition, certain frequencies inside this frequency band can be suppressed, for example to prevent interference with other systems. It is also possible to use two or more non-contiguous sub-bands for the transmission of a single data stream;
- each of the carriers can be modulated individually, with different modulation schemes, if appropriate. Typical examples of carrier modulation schemes are FSK, PSK, and QAM, with a different number of bits per carrier. With this flexible choice, the available signal to noise ratio can be used optimally for each carrier;

NOTE 1 The peak power required for a large number of carriers is about 10 dB higher than that of a single-carrier system. However, there are known ways to reduce the peak power of traditional MCM without affecting its performance.

- MCM is considerably more robust against intersymbol interference (ISI) or group delay distortion caused by the transmission channel than narrowband systems. This is mainly due to the fact that the parallel transmission on several carriers leads to a longer symbol duration. Furthermore, ISI can be completely eliminated by inserting guard intervals or a cyclic prefix between the symbols;
- MCM is robust in presence of narrowband interferers (continuous wave noise), because such jammers typically destroy only a single carrier. With proper forward error correction coding, the destroyed bits can be reconstructed;
- in combination with a well-designed interleaver and forward error correction coding scheme, MCM can be made robust against impulsive noise.

NOTE 2 This implies a more complex receiver structure, compared with, for example a simple FSK receiver, but the advantages listed above more than justify the use of MCM. There are FFT-based receiver structures whose complexity increases with $M \log_2 M$, where M is the number of carriers.

Due to the block processing of the MCM demodulator, an inherent transmission delay is introduced. However, for typical power line communication applications this delay is negligible.

### 5.2 Physical layer data format

#### 5.2.1 Purpose

This clause covers the services required for the PHY layer and the transmission methods which are used to provide the information flow through the physical channel (power distribution network).

#### 5.2.2 Transmission method, overview

This subclause specifies the transmission method of the MCM profile. The chosen modulation scheme is multicarrier differential phase shift keying with I carriers (IC-DPSK). The carrier frequencies are multiples of 4,5 kHz and the number I and the carrier frequencies are configurable. I bits per symbol are transmitted, leading to a gross data rate of I·4,5 kbit/s. To increase the robustness with respect to channel impairments, a rate 1/2 convolutional code is used and the length information and the integrity of a telegram are checked with cyclic redundancy check codes. The synchronization preamble assures a robust synchronization even in bad channel conditions.

To improve the performance in channels with a large group delay distortion, a cyclic prefix of configurable length can be used for the modulation of the payload. The synchronization preamble is always transmitted without cyclic prefix.

The data is transmitted with 288 k samples per second (64 samples per symbol[1]). In the receiver, the signal is sampled at 288 kHz and a 64 point FFT is performed.

[1] When a cyclic prefix is used, there are $N_{SS}$ samples per symbol.

It is assumed that the P_SDU Q(m), m = 0..8M–1, Q(m) ∈ [0,1] is to be transmitted with I bits/symbol using I subcarriers. The length information and the payload are each protected with a separate CRC. The resulting bit stream is padded and segmented into blocks which are interleaved and encoded. The data are then prepended by the synchronization sequence and modulated, see figure 3. A detailed description of each function is given below.

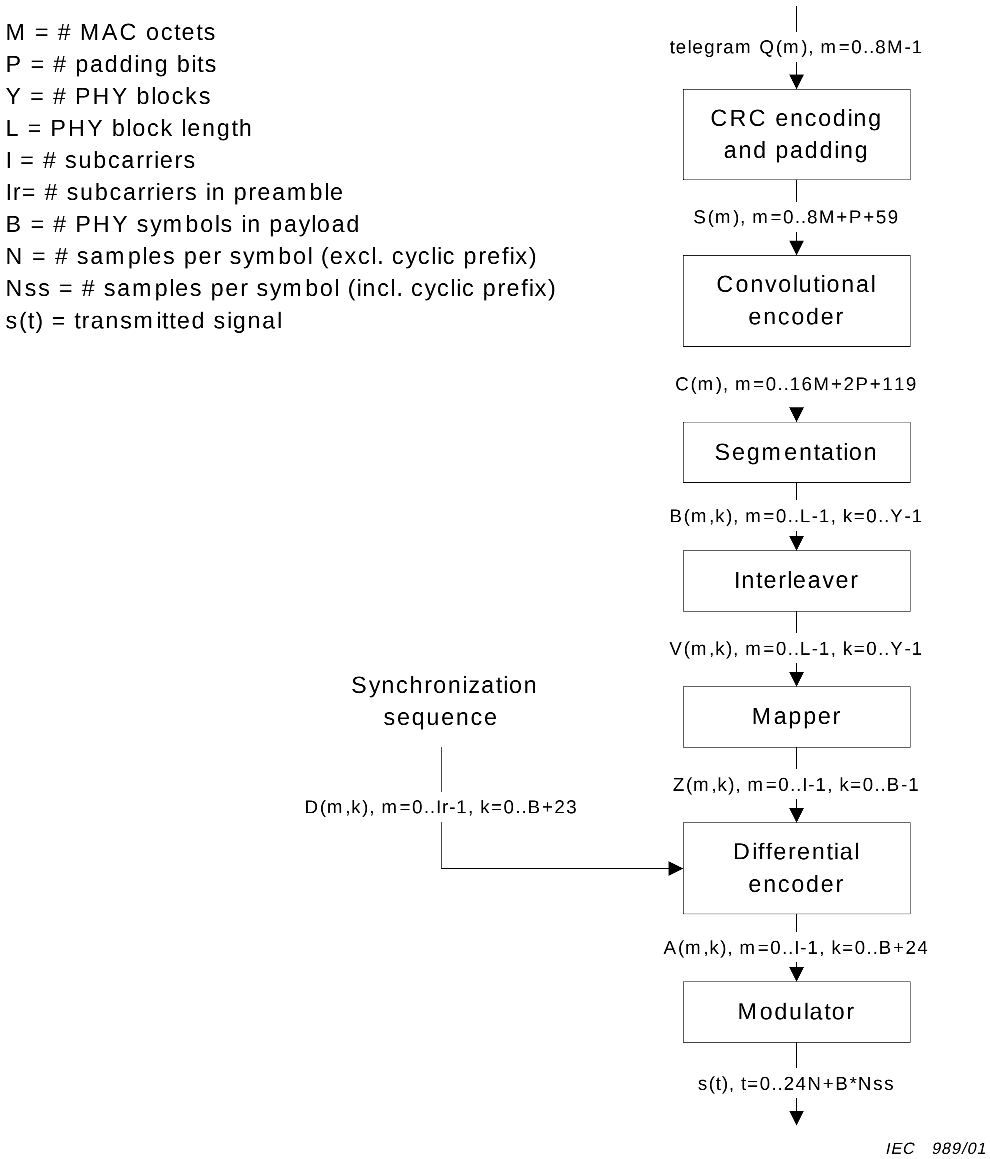



**Figure 3 – Transmitter data flow diagram (one telegram)**

### 5.2.3 Configuration parameters

The physical layer as described below is specified by the following design parameters, which can be configured in the network or adapted to the changing channel conditions. These parameters have to be identical in a network to achieve compatibility.

- *Number I of subcarriers* 1 ≤ I ≤ N/2–1. A typical value is N = 64.
- *Indices $i_1$ to $i_I$ of subcarriers* The subcarrier frequency is $i_x$·288 kHz /N, 1 ≤ $i_x$ ≤ N/2–1. This permits usage of non-contiguous frequency bands. Theoretical frequency range is from 288 kHz/N to 144–288/N kHz (i.e. excluding $i_x$ = 0 and $i_x$ = N/2). Practical frequency range to be chosen in accordance with IEC 61334-3-1.

- *Phases cos($\varphi_1$) and sin($\varphi_1$) to cos($\varphi_l$) and sin($\varphi_l$) of each subcarrier* $\varphi_x$ is the carrier phase of the subcarrier at index $i_x$. The phases can be chosen to reduce the peak power. Compatibility can be achieved even with different phases.
- *Usage flag $r_1$ to $r_l$ for preamble of each subcarrier* $r_x$ = 1 indicates that subcarrier x is to be used in the preamble. Otherwise, $r_x$ = 0.

NOTE This definition implies that the set of subcarriers used in the preamble is a subset of the subcarriers used in the payload.

- *Preamble phases cos($\varphi_{r1}$) and sin($\varphi_{r1}$) to cos($\varphi_{rl}$) and sin($\varphi_{rl}$) of each subcarrier* $\varphi_{rx}$ is the carrier phase of the subcarrier at index $i_x$ to be used in the preamble. $\varphi_{rx}$ is only meaningful for {x| $r_x$ = 1}. The phases can be chosen to reduce the peak power. Compatibility can be achieved even with different phases.
- *Length of cyclic prefix in samples, $N_{CP}$* This is introduced to cater for large group delay variations. Range is 0..63, default is 0.
- *Block length (L) in bits.* Blocks are defined in the segmentation process (see 5.2.6).

### 5.2.4 PHY PDU format, CRC encoding and padding

The PHY telegram structure is shown here,

| Field name | Preamble | LEN | RES | PAD_LEN | LEN_CRC | PL | PAD | PL CRC | FLUSH |
|---|---|---|---|---|---|---|---|---|---|
| Length (bits) | 25 | 8 | 8 | 8 | 16 | 8M | P | 16 | 4 |
| | Preamble | S(m) | | | | | | | |

Preamble and S(m) are encoded and modulated separately.

#### 5.2.4.1 Preamble

See below.

#### 5.2.4.2 LEN

LEN is the length of S in blocks: LEN = (8M + PAD LEN + 60)/BLOCK LEN

#### 5.2.4.3 RES

The RES field is reserved for future use. It shall contain 0 for the current version.

#### 5.2.4.4 PAD LEN

The PAD LEN field is the length P of the PAD field in bits.

#### 5.2.4.5 LEN CRC

The LEN CRC field $U_L(m)$, m = 0..15, contains the CRC checksum over the fields LEN, RES and PAD LEN. It is calculated as follows:

the remainder of the division of the polynomial $\sum_{m=0}^{23} S(m)x^m$ by the polynomial $X^{16} + X^{13} + X^{12} + X^{11} + X^{10} + X^8 + X^6 + X^5 + X^2 + 1$ is inverted and forms $U_L(m)$, $U_L(0)$ is the LSB.

NOTE This polynomial is taken from IEC 60870-5-1 for format class FT3. It represents an optimum BCH-code with Hamming distance 6 for block lengths ≤ 151 bits.

$U_L(m)$ is then inserted into S(m):

$S(m + 24) = U_L(m)$, m = 0..15

#### 5.2.4.6 PL

The payload field contains the PHY SDU of M bytes.

#### 5.2.4.7 PAD

The padding field is used to ensure that the encoded PHY telegram exactly fits into one or multiple PHY blocks, see below.

#### 5.2.4.8 PL CRC

The PL CRC field $U_{PL}(m)$, m = 0..15, contains the CRC checksum over the PL field. It is calculated as follows:

The remainder of the division of the polynomial $\sum_{m=0}^{8M-1} S(m+40)x^m$ by the polynomial $X^{16} + X^{13} + X^{12} + X^{11} + X^{10} + X^8 + X^6 + X^5 + X^2 + 1$ is inverted and forms $U_{PL}(m)$, $U_{PL}(0)$ is the LSB.

NOTE This polynomial is taken from IEC 60870-5-1 for format class FT3. It represents an optimum BCH-code with Hamming distance 6 for block lengths ≤ 151 bits.

$U_{PL}(m)$ is then inserted into S(m):

$S(m + 8M + 40 + P) = U_{PL}(m)$, m = 0..15

#### 5.2.4.9 FLUSH

The FLUSH field is used to flush the convolutional encoder, see below.

### 5.2.5 Convolutional encoding

The uncoded PHY telegram S(m), m = 0..8M + P + 59 is convolutionally encoded to form the encoded PHY telegram C(m), m = 0..16M + 2P + 119. The encoder is a rate 1/2 convolutional encoder with constraint length G = 5 and code generator “polynomials” 10111 and 11001. At the beginning, the encoder state is set to zero. The bit generated by the first code generator is output first. The use of the FLUSH field causes the encoder to be flushed, such that at the end the encoder is again in state zero. The block diagram of the encoder is shown in figure 4.

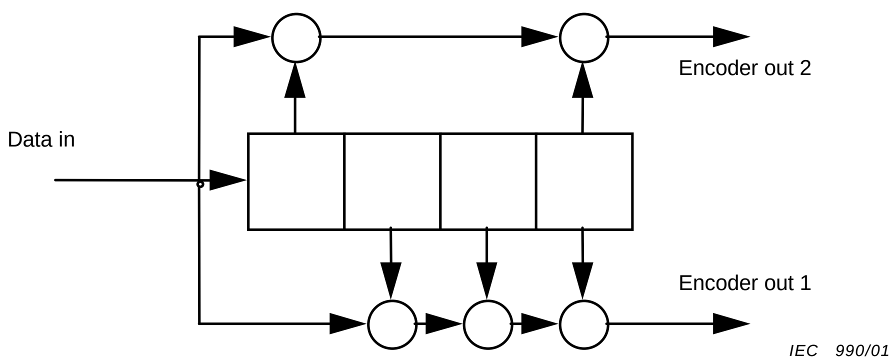



**Figure 4 – Block diagram of encoder**

### 5.2.6 Segmentation and interleaving

The encoded PHY telegram is segmented into PHY blocks over which intra-block interleaving is performed. The length of a PHY block, L, is a system parameter and has to be agreed upon in order to achieve compatibility between suppliers.

Since the length of the encoded PHY telegram C(m) is 16M + 120 + 2P, it can be segmented into Y PHY blocks of length L using P padding bits:

$$Y = \lceil (16M + 120)/L \rceil$$

$$P = (Y \cdot L - (16M + 120))/2$$

The segmentation into blocks B(m,k) is done as follows:

$$B(m,k) = C(m + k \cdot L), \qquad m = 0..L-1,\ k = 0..Y-1$$

The first index specifies the bit inside a block and the second index is the block number.

The block B(m,k) is transformed into a block V(m,k) using intra-block interleaving. The interleaving depends on the number I of subcarriers and on channel conditions and has to be agreed upon by the suppliers. Default is non-interleaving.

The resulting interleaved PHY telegram V(m,k) of size $L \cdot Y$ is mapped into Z(m,k) of size $I \cdot B$, where B is the number of symbols, $B = Y \cdot L/I$.

$$Z((m + k \cdot L) \bmod I,\ (m + k \cdot L) \operatorname{div} I) = V(m,k), \quad m = 0..L-1,\ k = 0..Y-1$$

Z(m,k) now contains the interleaved data to be transmitted. m = 0..I–1 denotes the carrier number, k = 0..B–1 the symbol number.

### 5.2.7 Preamble

The synch preamble consists of the sequence X(k) (sync preamble).

X(0..24) = 1 1 1 1 1 0 1 0 1 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1

The preamble is repeated in all subcarriers that are used in the preamble to form the burst D(m,k):

$$D(m,k) = X(k), k = 0..22, \qquad m = \{0..N/2-1 | r_m = 1\}$$

### 5.2.8 Modulation

The preamble D(m,k) is transmitted using multicarrier differential phase shift keying with $I_r$ subcarriers. The unmodulated PHY telegram Z(m,k) is modulated as a multicarrier differential phase shift keying (MC-DPSK) signal with I subcarriers and I bits per symbol.

First, the preamble D(m,k), $m = \{0..N/2-1 | r_m = 1\}$, k = 0..22 is differentially encoded in the time domain, yielding the differentially encoded preamble A(m,k), $m = \{0..N/2-1 | r_m = 1\}$, k = 0..23:

$$A(m,0) = 1, \qquad m = \{0..N/2-1 | r_m = 1\}$$

$$A(m,k) = 2(D(m,k-1) \oplus A(m,k-1)) - 1, \qquad m = \{0..N/2-1 | r_m = 1\},\ k = 1..23$$

where A(m,0) is the reference symbol for the preamble, and the symbol ⊕ represents modulo-2 addition.

The unmodulated PHY telegram Z(m,k) is then differentially encoded:

$A(m,24) = 1, \quad m = \{i_k\}, k = 1..I$

$A(m,k) = 2(Z(m,k-25) \oplus A(m,k-1))-1, \qquad k = 25..B + 24, m = \{i_k\}, k = 1..I$

where A(m,24) is the reference symbol for the payload.

Now, A(m,k) is a ternary signal to be transmitted on frequency k at time n. A(m,k) has values +1 for a binary ‘1’, –1 for a binary ‘0’ and 0 if no signal is to be transmitted.

Each symbol is modulated to form the signal s(t):

Preamble:

$$s(t) = \sum_{m=\{m|r_m=1\}} A\left(m,\left\lfloor \frac{t}{N} \right\rfloor\right)\cos(2\pi f_0 \cdot (i_m)t / N + \varphi_{rm}) \cdot p_1\left(t - N\left\lfloor \frac{t}{N} \right\rfloor\right) \quad t = 0..24N - 1/N$$

Payload:

For each symbol, generate a signal, using the data bits

$$\sum_{m=1}^{I} A\left(m,\left\lfloor \frac{t}{N} \right\rfloor\right)\cos(2\pi f_0 \cdot (i_m)t / N + \varphi_m) \cdot p_1\left(t - N\left\lfloor \frac{t}{N} \right\rfloor\right) \quad t = 0..N - 1$$

resulting in a vector of N = 64 samples. Copy of the last $N_{CP}$ samples of the 64-sample vector into an $N_{CP}$-sample vector. Prepend this $N_{CP}$-sample vector to the original 64-sample, resulting in an ($N_{CP}$ + 64)-sample symbol.

$p_1(t)$ is a rectangular pulse of length N:

$$p_1(t) = \begin{cases} 1 & t = 0..N - 1 \\ 0 & \text{otherwise} \end{cases}$$

$f_0$ = 288/N kHz. The bits shall be transmitted in increasing order of their numbering.

The preamble shall always be transmitted without cyclic prefix.

## 5.3 PHY services

### 5.3.1 PHY to MAC interface

#### 5.3.1.1 PHY_DATA.request

##### 5.3.1.1.1 Function

The PHY_DATA.request primitive is passed to the PHY layer entity to request that a PHY PDU be sent to one or several remote PHY entity or entities using the PHY transmission procedures.

#### 5.3.1.1.2 Structure

The semantics of this primitive are as follows:

PHY_DATA.request{
P_SDU}.

The P_SDU (PHY service data unit) parameter specifies the PHY service data unit to be transmitted by the PHY layer entity. There is sufficient information associated with P_SDU for the PHY sublayer entity to determine the length M of the data unit.

#### 5.3.1.1.3 Use

The primitive is generated by the MACphy sublayer entity whenever data is to be transmitted to a peer MAC entity or entities.

The receipt of this primitive will cause the PHY entity to perform all PHY specific actions (see 5.2.2) and pass the properly formed PDU to the mains attachment unit for transfer to the peer PHY layer entity or entities.

### 5.3.1.2 PHY_DATA.confirm

#### 5.3.1.2.1 Function

The PHY_DATA.confirm primitive has only local significance and provides an appropriate response to a PHY_DATA.request primitive. The PHY_DATA.confirm primitive tells the MAC sublayer entity whether the P_SDU of the previous PHY_DATA.request has been successfully transmitted.

#### 5.3.1.2.2 Structure

The semantics of this primitive are as follows:

PHY_DATA.confirm{
Result}.

The result parameter is used to pass status information back to the local requesting entity. It is used to indicate the success or failure of the previous associated PHY_DATA.request.

#### 5.3.1.2.3 Use

The primitive is generated in response to a PHY_DATA.request.

It is assumed that the MAC sublayer has sufficient information to associate the confirm with the corresponding request.

### 5.3.1.3 PHY_DATA.indication

#### 5.3.1.3.1 Function

This primitive defines the transfer of data from the PHY layer entity to the MAC sublayer entity.

#### 5.3.1.3.2 Structure

The semantics of this primitive are as follows:

PHY_DATA.indication{
P_SDU}.

The P_SDU parameter specifies the PHY service data unit as received by the local PHY sublayer entity.

#### 5.3.1.3.3 Use

The PHY_DATA.indication is passed from the PHY layer entity to the MAC sublayer entity to indicate the arrival of a valid PDU.

# 6 MAC sublayer protocol specification

## 6.1 Overview

### 6.1.1 MAC communication network architecture

The communication network is subdivided into domains. Each of these domains is organized around a master station. Slave stations in a domain are registered to that master station. Communication with a different master station is supported for test and network management.

Medium access within a domain is controlled by one station at a time. This station is called the 'initiator'.

Communication always involves the current 'initiator'. Direct communication between any two stations, of which neither is an 'initiator', is not supported.

### 6.1.2 Features of the MAC sublayer

The MAC sublayer exhibits the following features:

– both confirmed and unconfirmed transmission of PDUs;
– initiator controlled medium access;
– support of varying processing times in remote stations using different MAC service classes;
– multi-hop transmission (routing repetitions) transparent to MAC users;
– transmission error detecting capabilities for transmission failures on any hop level through cascaded timers.

## 6.2 Transmission procedures

The MAC sublayer provides different service classes. Dependent on the requested service class, different transmission procedures are used.

MAC service class 0 uses the postponed confirmation scheme (figure 5): the MAC sublayer generates the confirmation at the end of the transmission by the physical layer.

MAC service classes 1 and 2 use round-trip delayed confirmation (figure 6): the MAC sublayer starts at a timer as long as the round trip delay needed for the remote station to transmit an LLC frame back. MAC service class 1 sets the timer value to $T_{delay1}$, MAC service class 2 sets the timer value to $T_{delay2}$ (see 6.9).

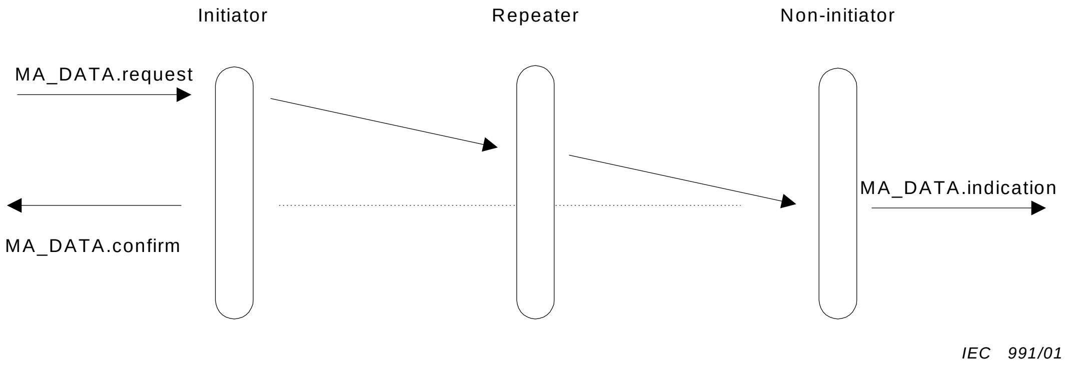



**Figure 5 – MAC transmission using MAC service class 0 (postponed confirmation)**

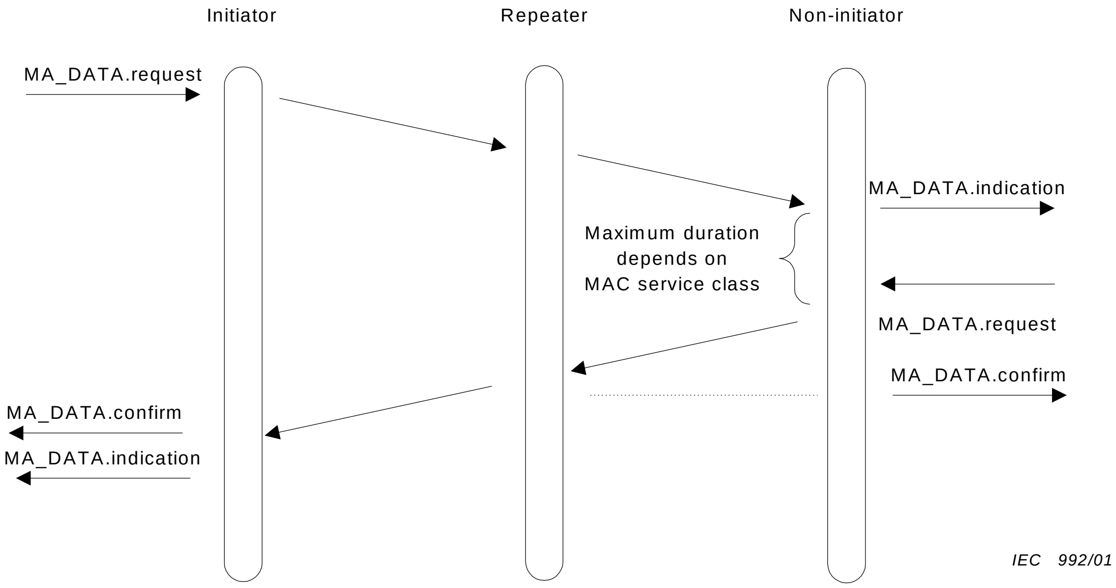



**Figure 6 – MAC transmission using MAC service class 1 or 2 (round-trip delayed confirmation)**

## 6.3 MAC services

### 6.3.1 MAC to LLC interface

#### 6.3.1.1 MA_DATA.request

##### 6.3.1.1.1 Function

The MA_DATA.request primitive is passed to the MACrore sublayer entity to request that a MAC SDU be sent to one or several remote MACrore entity or entities using the MACrore transmission procedures.

##### 6.3.1.1.2 Structure

The semantics of this primitive are as follows:

MA_DATA.request{

Destination_address,

M_SDU,

Service_class}.

The Destination_address parameter specifies an individual or group MAC address.

NOTE 1 A non-initiator MAC may only use an individual initiator MAC address as Destination_address.

The M_SDU (MAC service data unit) parameter specifies the MAC service data unit to be transmitted by the MAC sublayer entity. There is sufficient information associated with M_SDU for the MACrore sublayer entity to determine the length of the data unit.

The Service_class parameter specifies the type of service that the MACrore sublayer entity has to use to transmit the M_SDU. The parameter can take the following values which are associated with a certain MAC confirmation scheme:

| | |
|---|---|
| 0: | 'Postponed Confirmation, no reply from remote': The MAC sublayer postpones the confirmation until the complete transmission, including repetition steps, is carried out. Service class '0' is intended to be used with PDUs where no answer from the remote station(s) is allowed. |
| 1,2: | 'Round-trip delayed Confirmation 1,2': After transmission of a PDU, the MAC sublayer starts a timer with a value that takes the transmission time to the remote station (including processing time in intermediate repeaters), processing time in the remote station and transmission time of a reply (with a limited size) into account. After reception of a PDU from the remote MAC entity, the timer is stopped. At expiration of the timer, a MA_DATA.confirm with bad transmission status is generated. Service_class 1 and service_class 2 differ by the allowed processing time and the maximum size of the reply PDU. |

NOTE 2 The source MAC address is not specified since it is a local parameter that the MAC sublayer will fill in itself with regard to the protocol rules.

#### 6.3.1.1.3 Use

The primitive is generated by the LLC sublayer entity whenever data is to be transmitted to a peer LLC entity or entities.

The receipt of this primitive will cause the MAC entity to prepend all MAC specific fields (cf. MAC PDU description below) and pass the properly formed PDU to the lower layers of the protocol for transfer to the peer MAC sublayer entity or entities.

### 6.3.1.2 MA_DATA.confirm

#### 6.3.1.2.1 Function

The MA_DATA.confirm primitive has only local significance and provides an appropriate response to a MA_DATA.request primitive. The MA_DATA.confirm primitive tells the LLC sublayer entity whether the M_SDU of the previous MA_DATA.request could be transmitted.

#### 6.3.1.2.2 Structure

The semantics of this primitive are as follows:

MA_DATA.confirm{
    Transmission_status}.

The Transmission_status parameter is used to pass status information back to the local requesting entity. It is used to indicate the success or failure of the previous associated MA_DATA.request.

#### 6.3.1.2.3 Use

The primitive is generated in response to an MA_DATA.request. The conditions when to generate the MA_DATA.confirm primitive depends on the MAC service class of the previous MA_DATA.request primitive.

It is assumed that the LLC sublayer has sufficient information to associate the confirm with the corresponding request.

#### 6.3.1.3 MA_DATA.indication

##### 6.3.1.3.1 Function

This primitive defines the transfer of data from the MAC sublayer entity to the LLC sublayer entity.

##### 6.3.1.3.2 Structure

The semantics of this primitive are as follows:

```
MA_DATA.indication{
        Destination_address,
        Source_address,
        M_SDU}.
```

The Destination_address parameter specifies the destination address of the received MAC PDU.

The Source_address parameter is an individual MAC address as specified by the ADD field of the incoming PDU.

The M_SDU parameter specifies the MAC service data unit as received by the local MAC sublayer entity.

##### 6.3.1.3.3 Use

The MA_DATA.indication is passed from the MAC sublayer entity to the LLC sublayer entity to indicate the arrival of a valid PDU.

### 6.3.2 MAC layer management interface

#### 6.3.2.1 MA_EVENT.notify

##### 6.3.2.1.1 Function

This primitive defines the transfer of a detected event from the local MAC entity to the local layer management entity.

##### 6.3.2.1.2 Structure

The semantics of this primitive are as follows:

```
MA_EVENT.notify{
        Event_identifier
        Event_value1,
        Event_value2,
        Event_value3,
        Event_value4}
```

The Event_identifier identifies the type of event that was detected.

Event_value specifies additional event specific information.

##### 6.3.2.1.3 Use

This primitive is used to provide information about event occurrences at the local MAC sublayer.

#### 6.3.2.2 MA_SETMODE.request

##### 6.3.2.2.1 Function

The MA_SETMODE.request primitive is passed from the SMAP entity to the MACrore sublayer entity to set the mode of the MAC sublayer.

##### 6.3.2.2.2 Structure

The semantics of this primitive are as follows:

MA_SETMODE.request{
        Mode}.

The mode parameter specifies the operation mode according to which the MAC sublayer operates. It may take the values I_MODE if the MAC sublayer is requested to operate as initiator MAC and NI_MODE if the MAC sublayer is requested to operate as non-initiator MAC.

##### 6.3.2.2.3 Use

The primitive is generated by the SMAP entity when the operational mode of the data link layer is changed. The MA_SETMODE.request primitive does not result in the transmission of a MAC PDU.

#### 6.3.2.3 MA_SETMODE.confirm

##### 6.3.2.3.1 Function

The MA_SETMODE.confirm primitive is generated by the MAC sublayer in response to a previously issued MA_SETMODE.request. It indicates whether the MA_SETMODE.request was successful or not.

##### 6.3.2.3.2 Structure

The semantics of this primitive are as follows:

MA_SETMODE.confirm{
        Result}.

The result parameter indicates the success or failure of a previous mode setting command.

##### 6.3.2.3.3 Use

The primitive is generated in response to an MA_SETMODE.request.

#### 6.3.2.4 MIB variable access services

The MAC sublayer provides services to read and write the following MIB variables:

- routing path table;
- the station's own MAC address;
- timer values and additional variables for time-out calculation for different MAC service classes;
- maximum number of repeaters;
- size of domainID and nodeID field of MAC addresses;
- maximum size of non-initiator PDUs for transmission using MAC service class 1 or 2.

### 6.4 MAC PDU format

The MAC PDU has the following structure:

| | | |
|---|---|---|
| MACtrl | ADD | LLC_ PDU |

#### 6.4.1 MAC control field

The MACtrl field has the following structure:

| **Field name** | I/N bit | Protocol | Mcls | ARS | NoR |
|---|---|---|---|---|---|
| **Size in bits** | 1 | 3 | 2 | 3 | 3 |

##### 6.4.1.1 I/N bit

This field defines whether the PDU is an initiator PDU (from an initiator to one or several non-initiators) or a non-initiator PDU (from a non-initiator to an initiator). It is used for address decoding. The value of the bit has the following meaning:

I/N = '0': PDU originated from an initiator (initiator PDU);

I/N = '1': PDU originated from a non-initiator (non-initiator PDU).

##### 6.4.1.2 Protocol

The protocol field is intended to be used for protocol extensions, for example the identification of different addressing schemes or protocol versions.

##### 6.4.1.3 Mcls

The Mcls field defines the MAC service class that was requested in the MA_DATA.request. The following values are assigned:

Mcls = 0: MAC Service_class = '0';

Mcls = 1: MAC Service_class = '1';

Mcls = 2: MAC Service_class = '2';

Mcls > 2: reserved.

The use of the Mcls field is described below with the routing repetition procedures.

##### 6.4.1.4 ARS (actual repetition stage)

This field defines how many times the PDU was already routing repeated for initiator PDUs and how many times the PDU still has to be routing repeated for non-initiator PDUs. The field has a range from 0 to 7 and determines which of the following address fields are the current transmitter and receiver addresses. The use is described below in context with the address field.

##### 6.4.1.5 NoR (number of repetitions)

This field defines how many routing repeaters are involved in the transmission (i.e. part of their MAC address is included in the address field). The valid range is from 0 to 7. A value of 0 indicates that no routing repetitions are required, the address field thus only consists of the initiator MAC DomainID and the non-initiator MAC address.

### 6.4.2 Address field

The structure of MAC addresses is described below.

The address field is of variable length and contains identifiers for the initiator involved in the transmission, the consecutive routing repeaters and the end station.

MAC addresses consist of a domain ID and a node ID. The following rules apply to avoid transmission of redundant information:

– a transmission always involves an initiator and one or several non-initiators,
– MAC addresses of routing repeaters involved in the transmission have an unequivocal relationship to the initiator's domain ID.

It is thus sufficient to identify the domain ID of the involved initiator, the node IDs of the routing repeaters that identifies these given the initiator's domain ID, and the MAC address of the addressed non-initiator(s) involved in the data exchange.

The sequence of addresses in a PDU with two intermediate repeaters is the following:

| **Field name** | Initiator domain ID | REP1 node ID | REP2 node ID | END address (domain ID + node ID) |
|---|---|---|---|---|
| **Size in bits** | 16 | 8 | 8 | 24 |
| **ARS value** | | 0 | 1 | 2 |

Depending on the I/N bit, the PDU is either sent from the left address to the right address or vice versa.

The ARS and NoR elements in the MACtrl field identify the two valid MAC addresses for the current transmission. The involved stations are those with MAC addresses right and left of the address boundary with index ARS as indicated above.

For initiator PDUs (I/N bit = '0'), the sender of the PDU is the MAC address left of the boundary and the receiver is the MAC address right of the boundary; for non-initiator PDUs vice versa.

END address may be a group address for initiator PDUs. All other addresses are required to be individual addresses.

A routing repeater station increments the ARS value by one for initiator PDUs and decrements the ARS value by one for non-initiator PDUs.

The initial value of the ARS field is 0 for initiator PDUs and NoR for non-initiator PDUs.

### 6.4.3 LLC PDU

The LLC PDU field contains the information transmitted in the MAC PDU. It is copied from M_SDU parameter in the MA_DATA.request primitive and copied to the M_SDU parameter in the MA_DATA.indication primitive.

The LLC PDU is not always present.

### 6.5 MAC addresses

The address is hierarchically structured, the format is the following:

| **Address part** | Domain ID | Node ID |
|---|---|---|
| **Length in octets** | 2 | 1 |

The node ID part of a MAC address is used to identify a specific station within a domain.

The following tables 1 and 2 give overviews of pre-defined and individually assigned addresses:

**Table 1 – MAC domain IDs**

| **Domain ID** | **Used for** |
|---|---|
| <NoDomainID> | Pre-defined group domain ID that is not used for a domain. |
| <AllDomainsID> | Pre-defined group domain ID that is used as a group address for all domains. |
| <IndividualDomainID> | Individual domain ID used for an initiator and to construct MAC addresses of non-initiator stations. |

**Table 2 – MAC node IDs**

| **Node ID** | **Used for** |
|---|---|
| <NoBodyNodeID> | Node ID that is never recognized as own MAC address. |
| <InitiatorNodeID> | Individual node ID that is used to identify an initiator. |
| <IndividualNodeID> | Individual node ID that is used to identify a certain station within a domain |
| <ToAllNodeID> | Group node ID that is used as a group address for all stations which are part of a specified domain |

The following MAC addresses are pre-defined:

**Table 3 – MAC predefined addresses**

| | |
|---|---|
| <MyStations><br>domain ID = <IndividualDomainID><br>node ID = <ToAllNodeID> | Group address of all stations that are part of a certain domain |
| <AllStations><br>domain ID = <AllDomainsID><br>node ID = <ToAllNodeID> | Group address of all stations regardless of their state and the domain of which they are part |
| <NewStations><br>domain ID = <NoDomainID><br>node ID = <ToAllNodeID> | Group address of all stations not part of an individual domain |
| <NoStation><br>domain ID = <NoDomainID><br>node ID = <NoBodyNodeID> | Address never assigned to any station |

With respect to IEC 61334-4-1 (reference model of the communication), the predefined addresses in IEC 61334-4-1 correspond to:

**Table 4 – Mapping to IEC 61334-4-1 predefined MAC addresses**

| IEC 61334-4-1 | MCM profile MAC addresses |
|---|---|
| MASTERS (all initiators) | <AllDomainsID>, <InitiatorNodeID> |
| INITIATOR (in use initiator) | <IndividualDomainID>, <InitiatorNodeID> |
| "New" address | <NewStations> |
| "To All" address | <MyStations> |
| "To All Physical Stations" address | <AllStations> |
| Individual slave address | <IndividualDomainID>, <IndividualNodeID> |
| Individual initiator address | <IndividualDomainID>, <InitiatorNodeID> |
| NO-BODY | <NoStation> |



## 6.6 Used MAC PDUs

Two types of MAC PDUs may be given to the PHY layer for transmission. They are implicitly distinguishable through a void LLC PDU in the repetition control PDU.

### 6.6.1 Information PDU

– END_address = individual or group MAC address in initiator PDUs, individual MAC address in non-initiator PDUs;

– initiator domain ID = <IndividualDomainID>;

– LLC PDU not void.

This PDU is used to transmit information between LLC entities.

### 6.6.2 Repetition control PDU

– END address = individual MAC address;

– initiator domain ID = <IndividualDomainID>;

– LLC PDU void.

This PDU is used to report an error (time-out of an intermediate routing repeating station) to the initiator station (cf. routing repeater procedures below). END_address is the MAC address of the station that reports the repetition fault. The PDU may not be used as initiator PDU.

## 6.7 MAC invalid PDU

A MAC PDU is invalid when:

– the MACtrl field is not valid (e.g. ARS value larger than NoR, protocol field not valid).

## 6.8 MAC procedures

The MAC functions are:

– data transmission and receiving;

– LLC interfacing;

– address filtering when receiving a PDU;

– addressing;

– medium access;

– routing repetition of PDUs to final station.

These functions can be grouped in MACphy and MACrore functionality.

NOTE The exact interface between MACphy and MACrore is an implementation issue and is not specified here

### 6.8.1 MACphy procedures

#### 6.8.1.1 Transmit a PDU

When the MAC receives a request for the transmission of a PDU, MACphy passes the PDU to the PHY layer for transmission over the medium.

After transmission of a PDU, a timer with an appropriate time-out value is started. The time-out value depends both on the requested MAC service class and the routing path and is calculated by MACrore. MACrore will generate a MA_DATA.confirm at the latest at expiration of that timer.

#### 6.8.1.2 Receive a PDU

When a PDU is received, MACphy passes the PDU to the MACrore. Invalid MAC PDUs are discarded.

### 6.8.2 MACrore procedures

MACrore procedures are different depending on whether the local station acts as initiator, as routing repeater or as destination non-initiator station. Therefore, MACrore procedures are described by illustrating an example including one repeating station.

#### 6.8.2.1 Assembling of PDUs in reaction to MA_DATA.request

MACrore address generation in an initiator is different from that in a non-initiator.

In the initiator, address information is taken from the routing path table in the MIB. This table is maintained by the SMAP.

A non-initiator is only allowed to send PDUs to an individual initiator address. Routing information is taken from previously received PDUs.

#### 6.8.2.2 Routing repetition procedure

The following discussion of the routing repeater procedure is intended to serve as an example for better understanding of the basic idea. A complete description of the procedure is contained in the state tables.

Figure 7 shows the transmission of a PDU using MAC service class 1 or 2 with one involved routing repeater station.

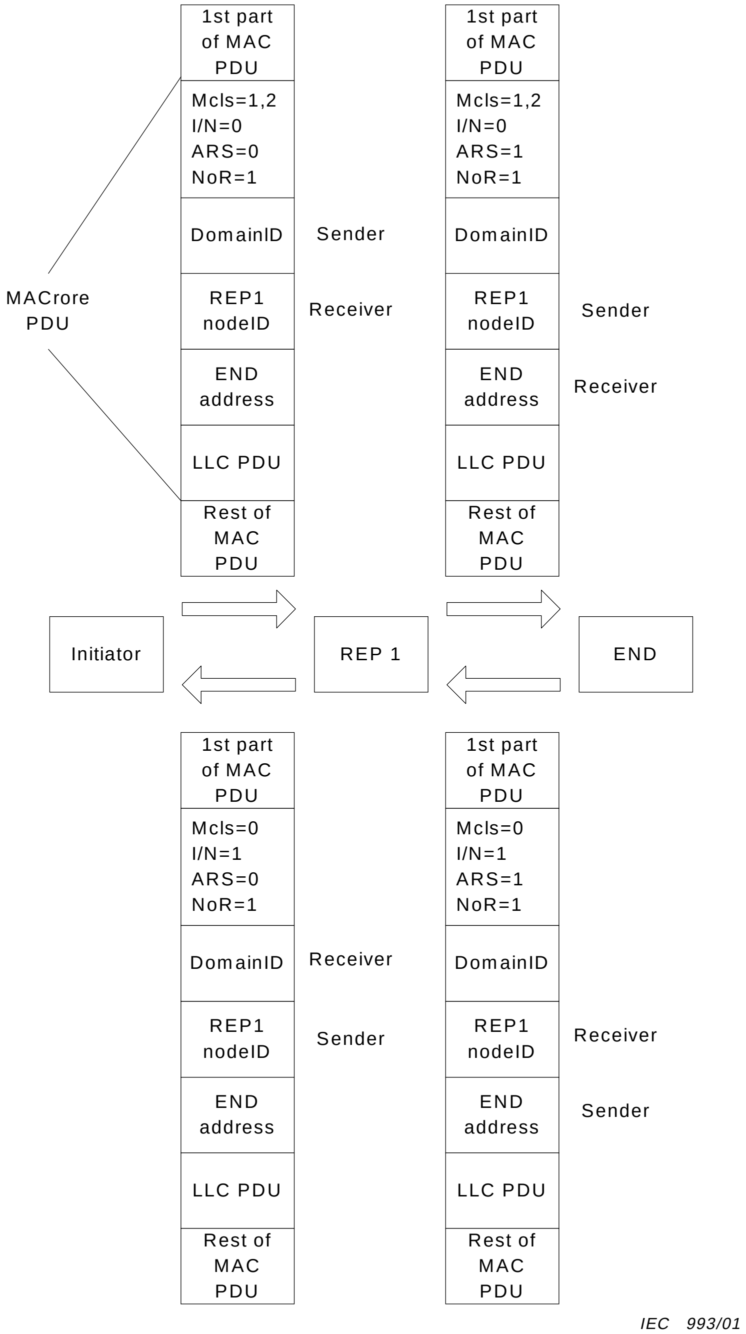



**Figure 7 – Example for transmission of a MAC PDU with one repetition using MAC service class 1 or 2**

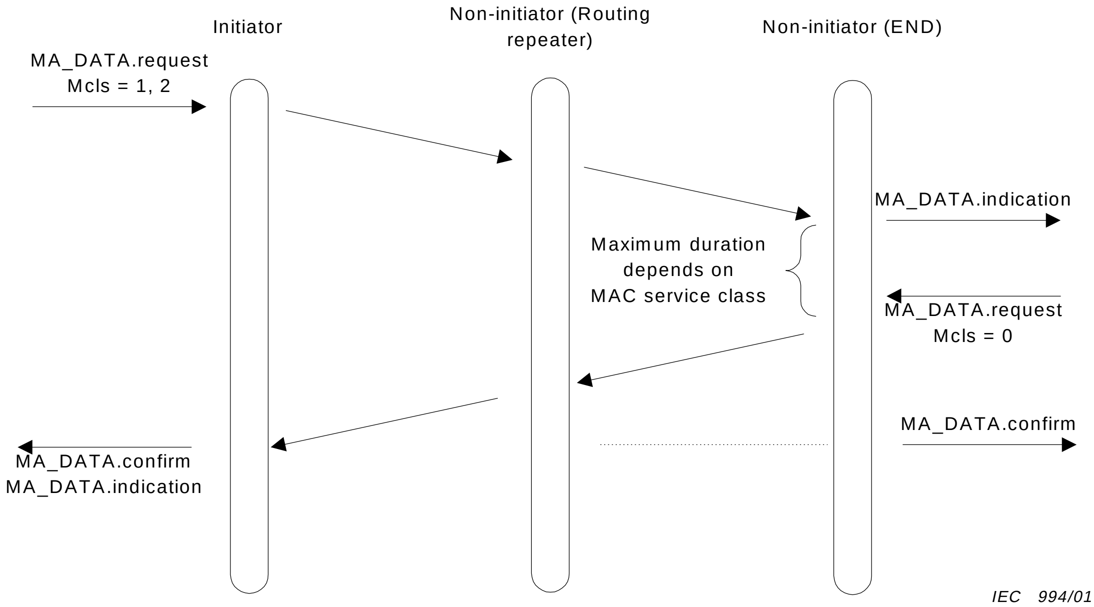



**Figure 8 – Time-sequence chart for (error-free) routing repeater procedure**

#### 6.8.2.2.1 Initiator procedures

The MACrore sublayer of the initiator examines the MA_DATA.request and taking into account:

– the initiator role;

– the requested end station address;

– the addresses of (possible) routing repeaters as extracted from the routing path table;

– the requested MAC service class.

builds a MACrore PDU with appropriate ADD and MACtrl fields and operates, dependent on the requested Service_class:

a) if the requested Service_class is 0: MACrore calculates the time-out value for timer T1 and forwards the PDU to MACphy. As soon as MACphy receives a positive PHY_DATA.confirm primitive, the timer is started. At expiration of T1 the MA_DATA.confirm primitive is issued to the requesting entity. The transmission is finished;

b) if the requested Service_class is 1-2: MACrore calculates the time-out value for timer T2 and forwards the PDU to MACphy. As soon as MACphy receives a positive PHY_DATA.confirm primitive, the timer is started. MACrore waits for reception of a PDU from the END-addressed non-initiator station or a repetition control PDU from one of the intermediate routing repeaters.

NOTE Under error-free protocol operation, the LLC of the END station will generate a MA_DATA.request in response to the received PDU.

If it receives a PDU before expiration of T2, it generates a MA_EVENT.notify with 1-way quality information. It examines the MACtrl and ADD fields and taking into account:

– the assigned initiator role;

– the ADD field of the received PDU in comparison to the ADD field of the previously issued PDU

stops timer T2 (if the address fields match) and generates a positive MA_DATA.confirm primitive for the previously issued MA_DATA.request and a MA_DATA.indication primitive with the received LLC PDU.

If the received PDU is a repetition control PDU, MACrore examines the end station address and generates an MA_DATA.confirm with appropriate error code.

If timer T2 expires without reception of any PDU, a MA_DATA.confirm with appropriate error code is generated and issued to the LLC entity.

#### 6.8.2.2.2 Routing repeater procedures

The MACrore sublayer of REP1 receives an initiator PDU, examines the MACtrl and ADD fields and taking into account:

– the assigned repeater role (I/N = 0, 0 ≤ ARS < NoR);

– the MAC Service_class

modifies the MACtrl field (increment ARS value by one) and operates according to the following procedure:

a) if Service_class is 0: the MAC sublayer sends the MAC PDU. The transmission is finished for the routing repeater;
b) if Service_class is 1-2: MACrore calculates the time-out value for timer T2 and forwards the PDU to MACphy. As soon as MACphy receives a positive PHY_DATA.confirm primitive, the timer is started. The routing information of the repeated PDU is stored as repetition control routing path because it is required for repetition failure reporting. Thereafter, MACrore waits for a non-initiator PDU from the addressed end station.

If it receives a PDU before expiration of T2, it examines the MACtrl and ADD fields and taking into account:

– the assigned repeater role (I/N = 1, 0 < ARS ≤ NoR)

stops T2, modifies the MACtrl field (decrements the ARS value by one) and passes the MACrore PDU to MACphy for transmission. The transmission is finished for REP1.

If timer T2 expires before receiving the non-initiator PDU, MACrore generates a repetition control PDU. The address field is taken from repetition control routing path. End address is the MAC address of REP1.

#### 6.8.2.2.3 End station procedures

The MACrore sublayer of the end station receives the initiator PDU, examines the MACtrl and ADD fields and taking into account the assigned end station role (ARS value = NoR value) generates a MA_DATA.indication primitive with the extracted LLC PDU at the LLC sublayer interface.

If the requested Service_class was 0, the transmission is finished.

If the Service_class is 1-2: MACrore calculates the time-out value for timer T2 and starts the timer. Thereafter, it waits for an initiator addressed reply to the received PDU. If it receives a MA_DATA.request before expiration of T2, it examines the MA_DATA.request and stops T2. Taking into account:

– the assigned end station role;

– the routing path of the previously received MAC PDU;

– the initiator address.

assembles a MAC PDU with an appropriate MACtrl field (Mcls = 0) and the ADD field containing the addresses of the routing path to the initiator. It passes the PDU to MACphy for transmission. As soon as MACphy gets a positive PHY_DATA.confirm, an appropriate MA_DATA.confirm primitive is generated. The transmission is finished.

If the timer T2 expires without the reception of a MA_DATA.request, a repetition control PDU is assembled and passed to the MACphy sublayer for transmission. Thereafter, the transmission is finished.

If a MA_DATA.request is received after expiration of T2, a MA_DATA.confirm with bad Transmission_status is generated.

### 6.9 MAC timer values

The respective timer values take into account:

| | |
|---|---|
| $T_{delay(1,2)}$ | The maximum response delay of the remote station (1 for service class 1, 2 for service class 2). These values are configurable system parameters and have to be agreed in order to achieve compatibility |
| $N_{rep}$ | The number of intermediate (remaining) routing repeaters to end station, extracted from the ADD field |
| $T_{rep}$ | The processing time in an intermediate routing repeater. This is a configurable system parameter and has to be agreed in order to achieve compatibility |
| $T_{spare}$ | Spare time increment per repetition stage to ensure proper fitting of the time-outs from the initiator and the intermediate routing repeaters |
| $T_{chan,\ cmd}$ | Transmission time for the PDU in command direction. This value depends on the size of the PDU, the number of used carriers for the physical layer, and the modulation method of the different carriers |
| $T_{chan,\ monmax}$ | Transmission time for the PDU in monitor direction. The maximum allowed size of the PDU has to be considered. This value depends on the number of used carriers for the physical layer, and the modulation method of the different carriers |

T1 for PDUs transmitted with MAC service class 0 is calculated as follows:

$$T1 = N_{rep} \cdot (T_{rep} + T_{chan,cmd}) \quad (1)$$

T2 for PDUs transmitted with MAC service class 1 or 2 is calculated as follows:

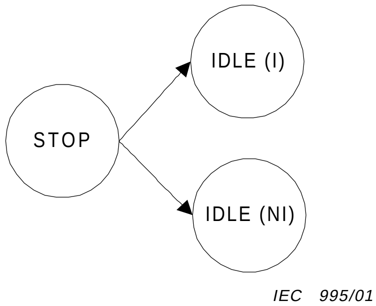
$$T2 = 2 \cdot N_{rep} \cdot T_{rep} + N_{rep} \cdot T_{chan,cmd} + (N_{rep} + 1) \cdot T_{chan,mon\,max} + T_{delay(1,2)} + T_{spare} \cdot N_{rep} \quad (2)$$

Some of the above values depend on the physical layer.

### 6.10 MAC state transition diagrams/tables

#### 6.10.1 Startup

##### 6.10.1.1 MAC startup state diagram




**I = initiator NI = non-initiator**

**Figure 9 – MAC startup state diagram**

#### 6.10.1.2 MAC startup state table

**Table 5 – MAC startup state table**

| State | Event | Condition | Action | Next state |
|---|---|---|---|---|
| STOP | MA_SETMODE.req | Mode = = I_MODE | set_mode(initiator)<br>MA_SETMODE.confirm(success) | IDLE (initiator) |
| STOP | MA_SETMODE.req | Mode = = NI_MODE | set_mode(non-initiator)<br>MA_SETMODE.confirm(success) | IDLE (non-initiator) |

#### 6.10.1.3 MAC startup state description

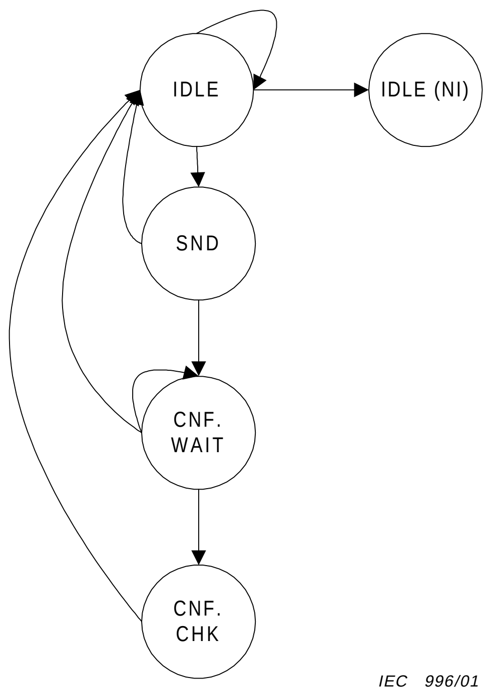
The MAC sublayer can be in one of the following states:

**Table 6 – MAC startup state description**

| | |
|---|---|
| STOP | MAC entity is waiting for a MA_SETMODE.req primitive |
| IDLE (initiator) | MAC entity is in IDLE state in initiator mode |
| IDLE (non-initiator) | MAC entity is in IDLE state in non-initiator mode |

### 6.10.2 MAC sublayer

#### 6.10.2.1 Initiator MAC state diagram




**NI = non-initiator**

**Figure 10 – Initiator MAC state diagram**

#### 6.10.2.2 Initiator MAC state table

**Table 7 – Initiator MAC state table**

| State | Event | Condition | Action | Next state |
|---|---|---|---|---|
| IDLE | MA_DATA.req | | path : = Get_path(dest_address)<br>Set_timer(Service_class, path)<br>Assemble_MACrore_PDU()<br>PHY_DATA.req () | SND |
| IDLE | PHY_DATA.ind | | MA_EVENT.notify() | IDLE |
| IDLE | MA_SETMODE.req | Mode = = NI_MODE | set_mode(non-initiator)<br>MA_SETMODE.confirm(success) | IDLE (non-initiator) |
| IDLE | MA_SETMODE.req | Mode = = I_MODE | MA_SETMODE.confirm(mode already set) | IDLE |
| NON-IDLE | MA_SETMODE.req | | MA_SETMODE.confirm(MAC busy) | = = |
| SND | PHY_DATA.con | result = = ok | Start_timer(Service_class) | CNF.WAIT |
| SND | PHY_DATA.con | result <> ok | MA_DATA.confirm(result) | IDLE |
| CNF.WAIT | Timeout | Service_class = = 0 | MA_DATA.confirm(success) | IDLE |
| CNF.WAIT | Timeout | Service_class > 0 | MA_DATA.confirm(repetition control error)<br>MA_EVENT.notify() | IDLE |
| CNF.WAIT | PHY_DATA.ind | MACphy_chk() <> ok | MA_EVENT.notify () | CNF.WAIT |
| CNF.WAIT | PHY_DATA.ind | MACphy_chk() = = ok | MACrore_chk() | CNF.CHK |
| CNF.CHK | | MACrore_add = = fits_sent<br>MACrore_frm = = info | Stop_timer()<br>MA_DATA.confirm(success)<br>MA_DATA.ind()<br>MA_EVENT.notify() | IDLE |
| CNF.CHK | | MACrore_add = = to_me<br>MACrore_frm = = rep_ctrl | Stop_timer<br>MA_DATA.confirm(repetition control error)<br>MA_EVENT.notify() | IDLE |
| CNF.CHK | | Else | MA_EVENT.notify() | CNF.WAIT |

#### 6.10.2.3 Initiator MAC state description

An initiator's MAC sublayer can be in one of the following states:

**Table 8 – Initiator MAC state description**

| | |
|---|---|
| IDLE | MAC entity is waiting for MA_DATA.request or PHY_DATA.indication primitives |
| SND | MAC entity requested transmission of a PDU. The MAC sublayer waits for local confirmation that the PDU was sent successfully |
| CNF | MAC entity is waiting for a condition to create a MA_DATA.confirm primitive. This is either a time-out or the reception of a requested reply. Subdivided into two states: |
| CNF.WAIT | Waiting for the condition to create MA_DATA.confirm |
| CNF.CHK | MAC entity checks whether address information matches previously received MA_DATA.request with Service_class > 0 |
| NON-IDLE | Any other state than IDLE |

#### 6.10.2.4 Initiator MAC procedures

**Table 9 – Initiator MAC procedures**

| | |
|---|---|
| Get_path (dest_address) | Extract routing path information from the routing path table. Default routing path is single hop |
| Set_timer (Service_class, path) | Set start value of the timer. The timer start values are calculated according to 6.9 |
| Assemble_MACrore_PDU () | Build the MACrore specific fields of a MAC PDU and pass the assembled PDU to MACphy for further processing |
| MACrore_chk () | Analyze the MACrore specific fields in the MAC PDU. Results of this procedure are the address information (source address, destination address) and the PDU type (information PDU or repetition control PDU) |
| Start_timer () | Start the timer. This procedure requires that the timer is set to an appropriate start value |
| Stop_timer () | Stop the timer before its expiration |
| MACphy_chk () | Perform the MACphy specific checks. Results of this procedure are the validation of a received PDU |
| PHY_DATA.req | Issue a PHY_DATA.request at the PHY layer interface |
| MA_DATA.con(result) | Generate MA_DATA.confirm primitive |
| MA_EVENT.notify() | Generate MA_EVENT.notify primitive |
| MA_DATA.ind() | Generate MA_DATA.indication primitive |
| MA_SETMODE.con() | Generate MA_SETMODE.confirm primitive |

#### 6.10.2.5 Initiator MAC variables

**Table 10 – Initiator MAC variables**

| | |
|---|---|
| MACrore_add | Address field of a received MACrore PDU. The variable is set as a result of MACrore_chk() |
| MACrore_frm | Frame type of a received MACrore PDU. The variable is set as a result of MACrore_chk() |
| Service_class | Requested MAC service class for an ongoing transmission sequence |
| path | Routing path information used to generate the MACtrl and ADD fields of a MACrore PDU and to calculate timer values |

#### 6.10.2.6 Initiator MAC conditions

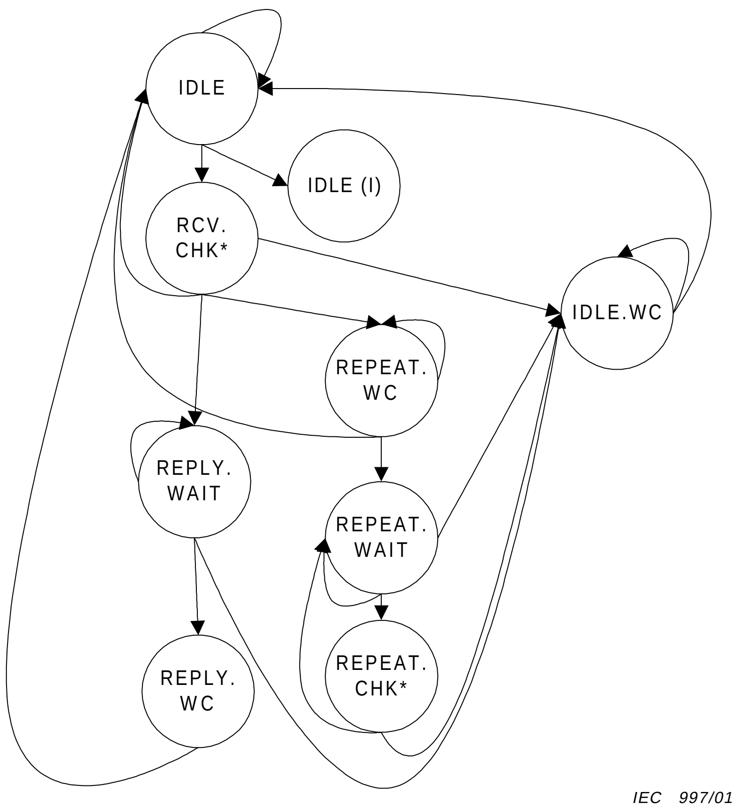
Most of the conditions are self-explanatory. In the following table, only those that require additional clarification are listed.

**Table 11 – Initiator MAC conditions**

| | |
|---|---|
| MACrore_add = = fits_sent | END_address is the same as the END address of the previously sent MAC PDU |
| MACrore_add = = to_me | The received MAC PDU is addressed to the initiator MAC address. If the MAC PDU is a repetition control PDU, the transmission is considered aborted by repetition control mechanisms |
| MACrore_frm = = info | The received MACrore PDU is an information PDU |
| MACrore_frm = = rep_ctrl | The received MACrore PDU is a repetition control PDU |

### 6.10.3 Non-initiator MAC sublayer

#### 6.10.3.1 Non-initiator MAC state diagram




**I = initiator**

**Figure 11 – Non-initiator MAC state diagram**

#### 6.10.3.2 Non-initiator MAC state table

The non-initiator state table is shown below.

**Table 12 – Non-initiator MAC state table**

| State | Event | Condition | Action | Next state |
|---|---|---|---|---|
| IDLE | PHY_DATA.ind | MACphy_chk<>ok | MA_EVENT.notify () | IDLE |
| IDLE | PHY_DATA.ind | MACphy_chk = = ok | MACrore_chk() | RCV.CHK* |
| IDLE | MA_SETMODE.req | Mode = = I_MODE | Set_mode(initiator)<br>MA_SETMODE.con(success) | IDLE (initiator) |
| IDLE | MA_SETMODE.req | Mode = = NI_MODE | MA_SETMODE.con(mode already set) | IDLE |
| IDLE | Other | | Default_actions(event) | IDLE |
| RCV.CHK* | | MACrore_add <> to_me<br>MACrore_add <> me_rep | MA_EVENT.notify () | IDLE |
| RCV.CHK* | | MACrore_add = = to_me<br>Service_class = = 0 | MA_DATA.ind()<br>MA_EVENT.notify () | IDLE |
| RCV.CHK* | | MACrore_add = = to_me<br>Service_class > 0 | MA_DATA.ind()<br>Save_path()<br>MA_EVENT.notify ()<br>Start_timer (Service_class, path, reply) | REPLY.WAIT |

| | | | | |
|---|---|---|---|---|
| RCV.CHK* | | MACrore_add = = me_rep<br>Service_class = = 0 | Modify_address()<br><br>PHY_DATA.req ()<br>MA_EVENT.notify () | IDLE.WC |
| RCV.CHK* | | MACrore_add = = me_rep<br>Service_class > 0 | Modify_address()<br><br>PHY_DATA.req ()<br>Save_path()<br>MA_EVENT.notify () | REPEAT.WC |
| REPLY.<br>WAIT | Timeout | | Generate_rep_ctrl<br><br>PHY_DATA.req ()<br>MA_EVENT.notify () | IDLE.WC |
| REPLY.<br>WAIT | MA_DATA.req | Destination = =<br>initiator(saved_path) | Stop_timer<br>Assemble_MACrore_PDU()<br><br>PHY_DATA.req () | REPLY.WC |
| REPLY.<br>WAIT | MA_DATA.req | Destination <><br>initiator(saved_path) | MA_DATA.con (medium<br>access denied) | REPLY.WAIT |
| REPLY.<br>WAIT | Other | | Default_actions(event) | REPLY.WAIT |
| IDLE.WC | PHY_DATA.con | | | IDLE |
| IDLE.WC | Other | | Default_actions(event) | IDLE.WC |
| REPLY.<br>WC | PHY_DATA.con | | MA_DATA.con(result) | IDLE |
| REPLY.<br>WC | Other | | Default_actions(event) | REPLY.WC |
| REPEAT.<br>WC | PHY_DATA.con | Result = = ok | Start_timer (Service_class,<br>path, repeater) | REPEAT.WAIT |
| REPEAT.<br>WC | PHY_DATA.con | Result <> ok | | IDLE |
| REPEAT.<br>WC | Other | | Default_actions(event) | REPEAT.WC |
| REPEAT.<br>WAIT | Timeout | | Generate_rep_ctrl()<br><br>PHY_DATA.req ()<br>MA_EVENT.notify () | IDLE.WC |
| REPEAT.<br>WAIT | PHY_DATA.ind | MACphy_chk<>ok | MA_EVENT.notify () | REPEAT.WAIT |
| REPEAT.<br>WAIT | PHY_DATA.ind | MACphy_chk = = ok | MACrore_chk() | REPEAT.<br>CHK* |
| REPEAT.<br>WAIT | Other | | Default_actions(event) | REPEAT.WAIT |
| REPEAT.<br>CHK* | | MACrore_add = = answer<br>OR<br>MACrore_add = =<br>RepCtrlPdu | Stop_timer()<br>Modify_address()<br><br>PHY_DATA.req ()<br>MA_EVENT.notify () | IDLE.WC |
| REPEAT.<br>CHK* | | MACrore_add <> answer | MA_EVENT.notify () | REPEAT.WAIT |

#### 6.10.3.3 Non-initiator MAC state description

A non-initiator's MAC sublayer can be in one of the following states:

**Table 13 – Non-initiator MAC state description**

| | |
|---|---|
| IDLE | MAC entity is waiting for MA_SETMODE.request or PHY_DATA.indication primitives |
| IDLE.WC | MAC entity is waiting for the PHY_DATA.con primitive in response to the transmission of a PDU that was not requested by the MAC user entity |
| RCV.CHK* | RCV.CHK is not an explicit state, but was introduced in the state table above to allow better readability. A received PDU is analyzed |
| REPEAT.CHK* | REPEAT.CHK is not an explicit state, but was introduced in the state table above to allow better readability. MAC entity received a PDU while in REPEAT.WAIT state and checks whether the PDU is the expected answer to a previously repeated initiator PDU |
| REPEAT.WAIT | MAC entity is waiting for the answer in monitor direction to a previously repeated initiator PDU with Service_class > 0 |
| REPEAT.WC | MAC entity is waiting for the PHY_DATA.con primitive in response to the transmission of a repeated initiator PDU |
| REPLY.WAIT | MAC entity is waiting for the LLC answer to a received PDU with Service_class > 0 |
| REPLY.WC | MAC entity is waiting for the PHY_DATA.con primitive in response to the transmission of a PDU previously requested by the MAC user entity |

### 6.10.3.4 Non-initiator MAC procedures

**Table 14 – Non-initiator MAC procedures**

| | |
|---|---|
| Assemble_MACrore_pdu () | Build the MACrore specific fields of a MAC PDU and pass the assembled PDU to MACphy for further processing. Path is taken from saved_path |
| Default_actions(event) | Perform default actions for the event:<br>MA_DATA.req → MA_DATA.con(medium access denied)<br>MA_SETMODE.req → MA_SETMODE.con(MAC busy)<br>PHY_DATA.ind → MA_EVENT.notify()<br>PHY_DATA.con → - |
| Generate_rep_ctrl() | Generate a repetition control PDU (MACrore PDU) to report the expiration of the reply/repetition timer to the current initiator. Path is taken from saved_path |
| MA_DATA.con(result) | Generate MA_DATA.confirm primitive |
| MA_DATA.ind () | Generate MA_DATA.indication primitive with the proper parameters |
| MA_EVENT.notify() | Generate MA_EVENT.notify primitive |
| MA_SETMODE.con(result) | Generate MA_SETMODE.confirm primitive |
| MACrore_chk () | Analyze the MACrore specific fields in the MAC PDU. Results of this procedure are the address information MACrore_add (source address, destination address) and the MAC service class of the received PDU |
| Modify_address() | Modify the address field (increment/decrement ARS field) according to the procedures described for routing repeater procedures |
| PHY_DATA.req() | Generate PHY_DATA.request primitive |
| Save_path() | Save the routing information of a received PDU in the variable saved_path |
| Set_mode() | Set the operational mode of the MAC sublayer to the requested mode |
| Start_timer(Service_class, path, reply/repeat) | Set start value of a timer and start the timer. The start value depends on the requested MAC service class and the station's role (i.e. END DLM or repeater with certain number of additional repeaters specified in the ADD field of the received PDU) |
| Stop_timer () | Stop the timer |

#### 6.10.3.5 Non-initiator MAC variables

**Table 15 – Non-initiator MAC variables**

| | |
|---|---|
| saved_path | Routing path information of a received telegram. Used to generate repetition control PDUs and to check received non-initiator PDUs while waiting for the response to a previously repeated initiator PDU |

#### 6.10.3.6 Non-initiator MAC conditions

Most of the conditions are self-explanatory. In the following table, only those that require additional clarification are listed.

**Table 16 – Non-initiator MAC conditions**

| | |
|---|---|
| MACrore_add = = me_rep | The MAC entity is used as a routing repeater |
| MACrore_add = = to_me | The received MAC PDU is addressed to the initiator MAC address. If the MAC PDU is a repetition control PDU, the transmission is considered aborted by repetition control mechanisms |
| MACrore_add = = RepCtrlPdu | The analysis of the address has as result that the received PDU is a repetition control PDU generated by a routing repeater MAC between the current station and the END addressed station |

**Standards Survey**

**The IEC would like to offer you the best quality standards possible. To make sure that we continue to meet your needs, your feedback is essential. Would you please take a minute to answer the questions overleaf and fax them to us at +41 22 919 03 00 or mail them to the address below. Thank you!**

Customer Service Centre (CSC)
**International Electrotechnical Commission**
3, rue de Varembé
1211 Genève 20
Switzerland

or

Fax to: **IEC**/CSC at +41 22 919 03 00

Thank you for your contribution to the standards-making process.

Nicht frankieren
Ne pas affranchir

Non affrancare
No stamp required

**A Prioritaire**

**RÉPONSE PAYÉE**

**SUISSE**

Customer Service Centre (CSC)
**International Electrotechnical Commission**
3, rue de Varembé
1211 GENEVA 20
Switzerland

**Q1** Please report on **ONE STANDARD** and **ONE STANDARD ONLY**. Enter the exact number of the standard: *(e.g. 60601-1-1)*

.............................................................

**Q2** Please tell us in what capacity(ies) you bought the standard *(tick all that apply)*. I am the/a:

- purchasing agent ❑
- librarian ❑
- researcher ❑
- design engineer ❑
- safety engineer ❑
- testing engineer ❑
- marketing specialist ❑
- other.....................................................

**Q3** I work for/in/as a: *(tick all that apply)*

- manufacturing ❑
- consultant ❑
- government ❑
- test/certification facility ❑
- public utility ❑
- education ❑
- military ❑
- other.....................................................

**Q4** This standard will be used for: *(tick all that apply)*

- general reference ❑
- product research ❑
- product design/development ❑
- specifications ❑
- tenders ❑
- quality assessment ❑
- certification ❑
- technical documentation ❑
- thesis ❑
- manufacturing ❑
- other.....................................................

**Q5** This standard meets my needs: *(tick one)*

- not at all ❑
- nearly ❑
- fairly well ❑
- exactly ❑

**Q6** If you ticked NOT AT ALL in Question 5 the reason is: *(tick all that apply)*

- standard is out of date ❑
- standard is incomplete ❑
- standard is too academic ❑
- standard is too superficial ❑
- title is misleading ❑
- I made the wrong choice ❑
- other ....................................................

**Q7** Please assess the standard in the following categories, using the numbers:
(1) unacceptable,
(2) below average,
(3) average,
(4) above average,
(5) exceptional,
(6) not applicable

- timeliness .............................................
- quality of writing....................................
- technical contents.................................
- logic of arrangement of contents ..........
- tables, charts, graphs, figures...............
- other ....................................................

**Q8** I read/use the: *(tick one)*

- French text only ❑
- English text only ❑
- both English and French texts ❑

**Q9** Please share any comment on any aspect of the IEC that you would like us to know:

............................................................
............................................................
............................................................
............................................................
............................................................
............................................................
............................................................
............................................................
............................................................
............................................................
............................................................
............................................................

ISBN 2-8318-5868-2

**ICS 33.200**

Typeset and printed by the IEC Central Office
GENEVA, SWITZERLAND