PD CLC/TS 50590:2015

BSI Standards Publication

Electricity metering data exchange — Lower layer PLC profile using Adaptive Multi Carrier Spread-Spectrum (AMC-SS) modulation

bsi.

...making excellence a habit.™

National foreword

This Published Document is the UK implementation of CLC/TS 50590:2015.

The UK participation in its preparation was entrusted to Technical Committee PEL/13, Electricity Meters.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Published by BSI Standards Limited 2015

ISBN 978 0 580 85918 2
ICS 35.240.60; 91.140.50

Compliance with a British Standard cannot confer immunity from legal obligations.

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 31 May 2015.

Amendments/corrigenda issued since publication

Date	Text affected

TECHNICAL SPECIFICATION
SPÉCIFICATION TECHNIQUE
TECHNISCHE SPEZIFIKATION

CLC/TS 50590

April 2015

ICS 35.240.60; 91.140.50

English Version

Electricity metering data exchange - Lower layer PLC profile using Adaptive Multi Carrier Spread-Spectrum (AMC-SS) modulation

This Technical Specification was approved by CENELEC on 2014-11-11.

CENELEC members are required to announce the existence of this TS in the same way as for an EN and to make the TS available promptly at national level in an appropriate form. It is permissible to keep conflicting national standards in force.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

CONTENTS

TABLE OF FIGURES

Foreword

This document (CLC/TS 50590:2015) has been prepared by CLC/TC 13 "Electrical energy measurement and control".

The following date is fixed:

- latest date by which the existence of this document has to be announced at national level (doa) 2015-07-24

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC [and/or CEN] shall not be held responsible for identifying any or all such patent rights.

This document has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association.

1 Scope

This Technical Specification specifies the physical layer, medium access control layer and logical link control layer for communication on an electrical distribution network between a master node and one or more slave nodes using a compatibly extendable form (CX1) of Adaptive Multi-Carrier Spread Spectrum (AMC-SS) technique. The adaptive cellular communication network technology provided in this specification may be used for automated meter reading as well as for other distribution network applications.

The[GK1] physical layer provides a modulation technique that efficiently utilizes the allowed bandwidth within the CENELEC A band (3 kHz – 95 kHz), offering a very robust communication in the presence of narrowband interference, impulsive noise, and frequency selective attenuation. The physical layer of AMC-SS is defined in Clause 5 of CLC/TS 50590:2015[GK2].

The data link (DL) layer consists of three parts, the ‘Medium Access Control’ (MAC) sub-layer, the Logical Link Control (LLC) sub-layer and the ‘Convergence’ sub-layer. The data link layer allows the transmission of data frames through the use of the power line physical channel. It provides data services, frame integrity control, routing, registration, multiple access, and cell change functionality. The MAC sub-layer and the LLC sub-layer of AMC-SS are defined in Clause 6 of CLC/TS 50590:2015. The Convergence sub-layer is defined in this document.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

EN 50065-1, *Signaling on low-voltage electrical installations in the frequency range 3 kHz to 148,5 kHz – Part 1: General requirements, frequency bands and electromagnetic disturbances*

DIN 43863–5:2012-04, *Identification number for measuring devices applying for all manufacturers*

3 Terms, definitions and acronyms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1
device identifier
property that universally identifies a node

3.1.2
frame forwarding
procedure of PHY frame retransmission by a slave node or simultaneously by several slave nodes

3.1.3
higher layer entity
entity of the layer or sub-layer in OSI model, which is situated above the layer or sub-layer of the entity offering the services. A possible convergence sub-layer between the higher layer entity and the entity offering the services may be a null layer, which is as simple as possible without a special convergence capability

3.1.4
MAC frame
MAC sub-layer Protocol Data Unit (MPDU)

3.1.5
master node
node which controls and manages the resources of a network cell

3.1.6
message
LLC sub-layer Protocol Data Unit (MPDU)

3.1.7
network
set of network nodes that can communicate by complying with this specification and are identified by the same value of N_NIN

3.1.8
network cell
set of network nodes that can communicate by complying with this specification and share a single master node, which is identified by the CIN

3.1.9
node
any one element of a network cell which is able to transmit to and receive from other network elements

3.1.10
slave node
any node of a network cell which is not operating as a master node

3.1.11
PHY frame
a PHY Layer Protocol Data Unit (PPDU)

3.1.12
registration
procedure of master node assignment to a slave node

3.1.13
slave node registration
assignment of dynamic address information to a slave node

3.1.14
symbol
waveform used in the communication channel that persists for a fixed period of time

3.2 Acronyms

AC	Alternating Current
ACK	Acknowledgement
AGC	Automatic Gain Control
AMC-SS	Adaptive Multi-Carrier Spread-Spectrum
BNC	Backbone Network Connection
BTO	Beacon Time-Out Period
CCV	Code Connection Vector
CENELEC	European Committee for Electrotechnical Standardization
CFL	Carrier Frequency List
CIN	Channel Identification Number
CL	Convergence Sub-Layer
CONV	Convolutional Code
COSEM	Companion Specification for Energy Metering
CRC	Cyclic Redundancy Check
CRCINITVAL	CRC Initialization Value
CWD	Contention Window Duration
CX1	Compatibly extendable form [JK3]of AMC-SS PLC
D_ACK	Data Acknowledgement
DB	Data Block
DBL	Data Block Length
DFC	Data Flow Control
D8PSK	Differential Eight Phase Shift Keying
DBPSK	Differential Binary Phase Shift Keying
DID	Device Identifier
DL	Data Link
DLL	Data Link Layer
DLMS	Device Language Message Specification
DLS	Data Link Services
DP	Data Priority
DPSK	Differential Phase Shift Keying
DQPSK	Differential Quadrature Phase Shift Keying
FCB	Frame Count Bit
FCS	Frame Check Sequence
FCV	Frame Count Bit Valid
FEC	Forward Error Correction
FH	Frequency Hopping
FHCS	Frame Header Check Sequence
FMS	Frequency mapping schemes
FX	Frame Forwarding
FXDC	Frame Forwarding Down-Counter
FXS	Frame Forwarding Sector Address
FXT	Total Number of Frame Retransmissions
HLE	Higher Layer Entity
HTB	Header Tail Bits
Hz	Hertz
IEC	International Electrotechnical Commission
IFI	Interframe Interval
ITD	Initial Transmission Delay
kHz	kilo Hertz
L_FC	Link Function Code
L_NIN	Lower Byte of Network Identification Number
LA	Link Address
LCN	Link Channel Number
LLC	Logical Link Control
LLCF	Logical Link Control Field
LPDU	LLC Sub-Layer Protocol Data Unit
LSB	Least Significant Byte
LTS	Length of Training Sequence

N_NIN	Node Network Identification Number
M_CIN	Master Channel Identification Number
MAC	Medium Access Control
MA	Multicast Link-Address
MCN	Multicast Number
MN	Master Node
MPDU	MAC Protocol Data Unit
MSB	Most Significant Byte
NIN	Network Identification Number
NLA	New Link Address
NMAx	New Multicast Addresses
NSC	Network with Spreading/Shrinking Cells
p-MN	Primary Master Node
OSI	Open System Interconnection
PDDTH	Power-Down Duration Threshold
PDS	PHY Data Symbol
PDU	Protocol Data Unit
PDZ	PHY Data Zero Symbol
PHS	PHY Header Symbol
PHY	Physical Layer
PHZ	PHY Header Zero Symbol
PPDU	PHY Protocol Data Unit
PRM	Primary Bit
PRMB	Preamble
PSDU	PHY Service Data Unit
PSK	Phase Shift Keying
RC	Repetition Code
RES	Reserved
RN	Relaying Node
RZCO	Reference Zero-Crossing Offset
s-MN	Secondary Master Node
SCA	Scrambling Code Array
SCL	Scrambling Code Length
SN	Slave Node
SYNC	Synchronization Sequence
SYNCR	Synchronization Sequence Reference Symbols
SYNCS	Synchronization Sequence Symbols
S_CIN	Slave Channel Identification Number
S_FXENA	Slave Frame Forwarding Enable
S_FXS	Slave Frame Forwarding Sector
S_MAx	Slave Multicast Addresses
S_LA	Slave Link-Address
TB	Tail Bits
TLA	Temporary Link Address
TM	Transmit Mode
TNCW	Total Number of Contention Windows
TNS	Total Number of Symbols
TS	Training Sequence
TSS	Training Sequence Symbol
TSA	Training Sequence Array
TSL	Training Sequence Length

4 General description

Layers 1 and 2 transport the higher layer messages between the nodes of a low voltage distribution network. Layer 1 (physical layer, PHY) generates a physical signal that is sent over the medium. The data link layer (DLL, layer 2) is split up into three sub-layers: medium access control sub-layer (MAC, sub-layer 2a), logical link control sub-layer

(LLC, sub-layer 2b) and convergence sub-layer (CL, sub-layer 2c). The first two sub-layers of the data link layer perform the formatting of the frames, handle channel access and frame forwarding procedures, provide data integrity checks and are responsible for addressing, segmentation and message retransmission. The convergence layer provides adaptation to the specific higher layer protocol and may be transparent. The convergence layer is not part of this document but is defined in the profile specification. The convergence sub-layer provides a mapping between the primitives that are used by the higher layer entity, and the primitives of the logical link control sub-layer. Layer 2 also provides functionality of multiplexing and prioritization between different higher layer entities or applications (within a network node) and layer-2 networking. Furthermore, multiplexing of different protocol elements with DLMS/COSEM elements is specified herein. The structure of the lower layer PLC profile is shown in Figure 1.

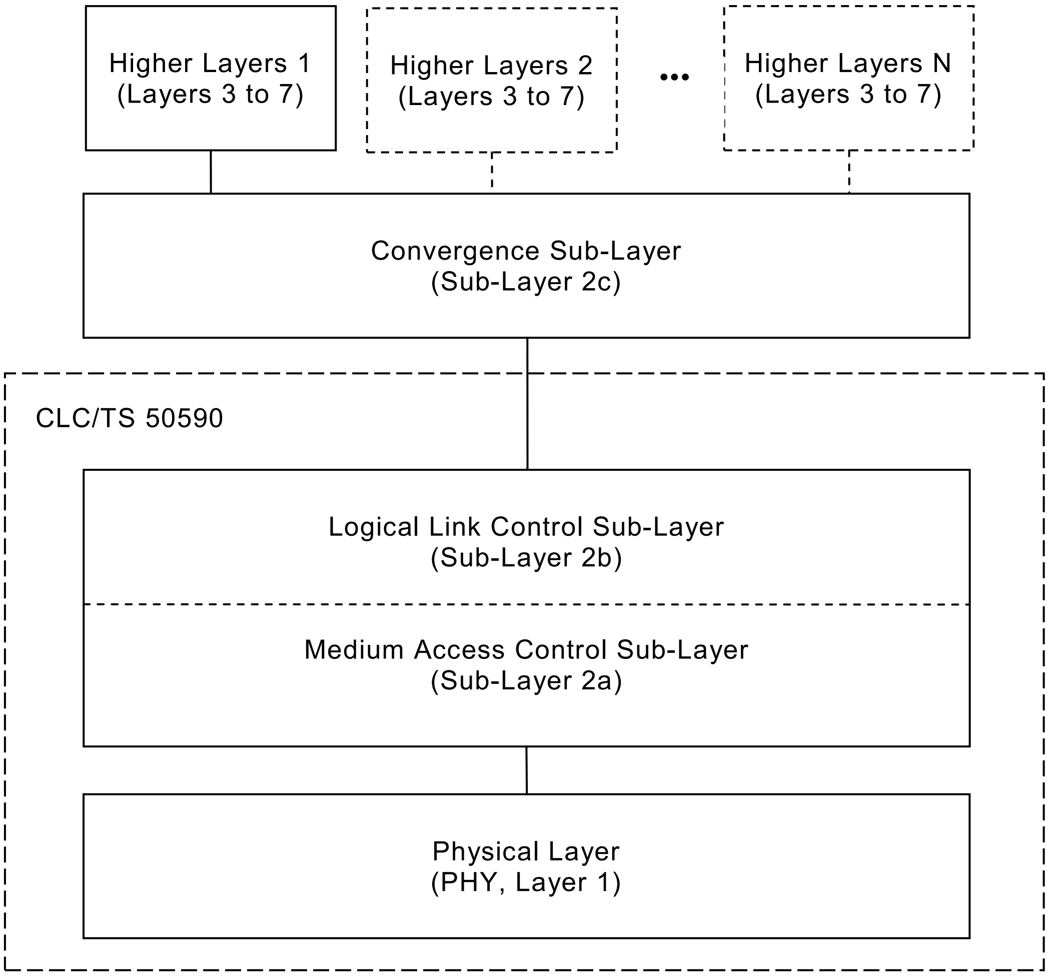

Figure 1 – Layers of AMC-SS profile

Information between layers is exchanged via primitives (see Figure 2). The following primitives are used for the communication between the logical link control sub-layer and the convergence sub-layer: DL_data.request, DL_data_identifier.confirm, DL_data.indication, DL_data.response, DL_data.confirm, DL_data_ack.response, DL_data_ack.confirm and DL_control.indication. For communication between the data link layer (LLC + MAC) and the physical layer the primitives P_data.request and P_data.indication are used.

.

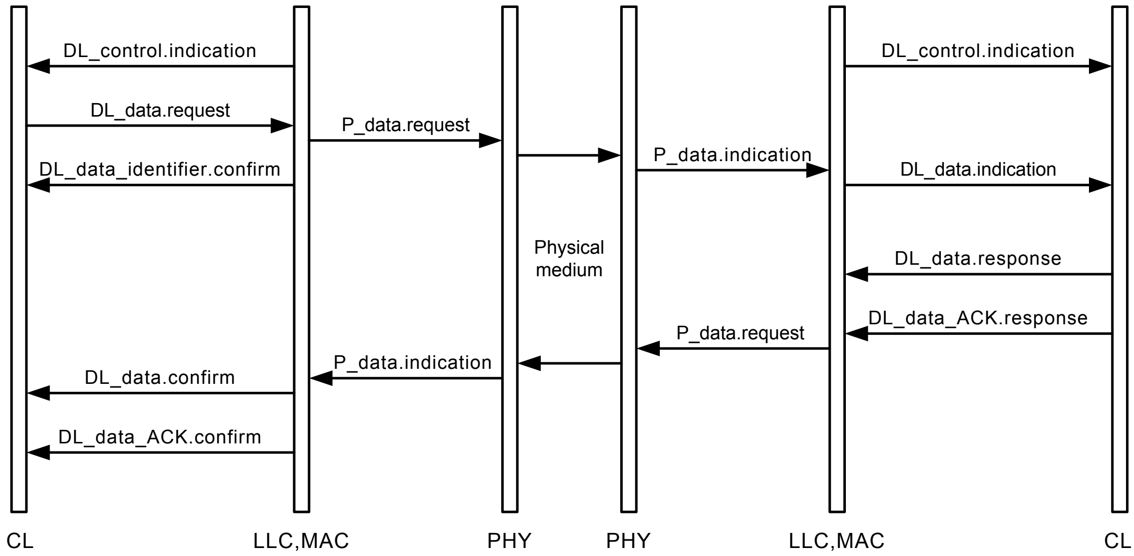

Figure 2 – Primitives

5 PHY layer specification

5.1 Overview

To generate physical signals, the AMC-SS physical layer uses a multi-carrier spread spectrum technique in combination with Differential Phase Shift Keying (DPSK) and forward-error-correction (FEC) coding.

This technique provides the following advantages:

- Robustness against time–frequency-selective fading;
- Robustness against pulse and narrowband interference, pulsating non-gaussian noise and combinations of them;
- Robustness against unwanted intermodulation effects;
- Lower linearity requirements for the analogue front end;
- High power efficiency as a result of low peak-to-average power ratio of the transmitted signal;
- Good electromagnetic compatibility between neighbouring systems.

The Figure 3 shows the block diagram of different data processing steps performed by the physical layer during the transmission of a PHY protocol data unit (PPDU).

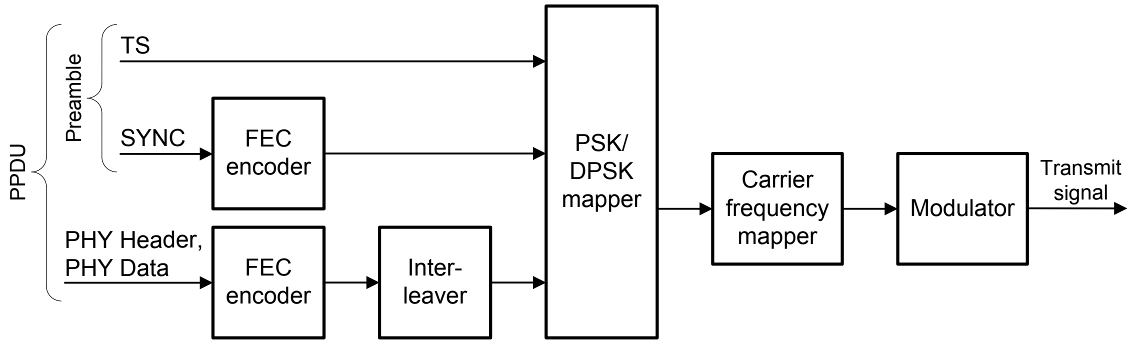

Figure 3 – PHY layer processing steps during PPDU transmission

Neither FEC-encoding nor interleaving is performed on the training sequence (TS). The bits of the synchronisation sequence (SYNC), the PHY header and PHY data are encoded with a FEC code. Encoded bit-sequences of the PHY header and PHY data are additionally interleaved. The training sequence and the encoded and interleaved bit sequences are mapped to the carrier phase angels depending on a PSK or a DPSK scheme and then to the carrier frequencies. The result of the PSK/DPSK and carrier frequency mapping is a sequence of unmodulated symbols containing modulation parameter. In the modulator these symbols are modulated and combined to a transmit signal, which is coupled into the physical channel – power-line.

The PHY layer processing steps during the reception of the PPDU are implementation specific and out of scope of this specification.

5.2 PHY protocol data unit

5.2.1 PPDU structure

5.2.1.1 General

The PPDU (i. e. PHY frame) consists of the preamble, the PHY header and the PHY data field. The preamble contains a training sequence (TS) and the synchronisation sequence SYNC. The PSK encoded elements of the TS are defined by the parameter TSL and the parameter array TSA (see Annex D).

Figure 4 below shows the bit-oriented structure of the PPDU (except training sequence), which is defined in Table 1 below.

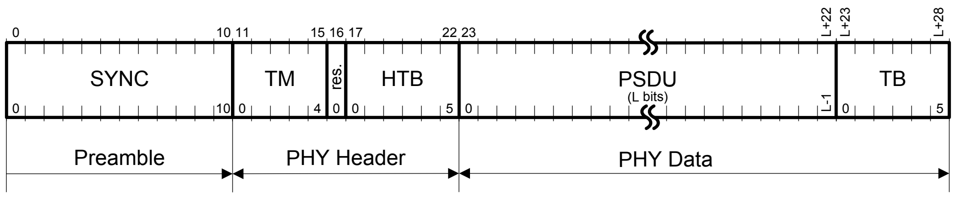

Figure 4 – Bit-oriented PPDU structure without TS

The bits of the PPDU (except TS) are numbered from 0 to L+28, where L is the length of PSDU in bits. In the following these bits are denoted as d_0, d_1, …, d_k, …, d_{L+28} .

Table 1 – PPDU structure without TS

		Field	Bit number	Bits	Name	Value
Part of PPDU	Preamble	SYNC	10 to 0	11	Synchronisation bit sequence	0x247 - allowed value; 0x000 to 0x246 - unused; 0x248 to 0x7FF - unused.
	PHY Header	TM [a]	15 to 11	5	PHY data transmit mode number [a]	0x00 to. 0x14 - TM0 to TM20; 0x15 to 0x1F – reserved for future use.
		res.	16	1	Reserved bit for future use	0x0 - allowed value; 0x1 - not allowed.
		HTB	22 to 17	6	PHY header tail bits	0x00 - allowed value; 0x01 to 0x3F - not allowed.
	PHY Data	PSDU	23 to L+22	L [a]	PHY service data unit	PSDU = MPDU (cf. Table 9).
		TB [b]	L+23 to L+28	6	PHY data tail bits [b]	0x00 - allowed value; 0x01 to 0x3F - not allowed.

[a] L is the length of PSDU in bits. The value of L is handed over to the PHY layer from MAC sub-layer together with MPDU = PSDU and the value of TM using the primitive P_data.request defined in Sub-section 5.5.

[b] PPDU does not contain TB if TM = 16 to 20

For the transmission of the TS, SYNC, PHY header and PHY data different transmit modes are used (see Table 2 below).

Figure 5 shows symbolically the transmit signal (corresponding transmitted PHY frame) consisting of overlapped modulated symbols at the output of the modulator (cf. Figure 7 and Figure 8). The detailed PHY frame structure is described in the following sub-sections.

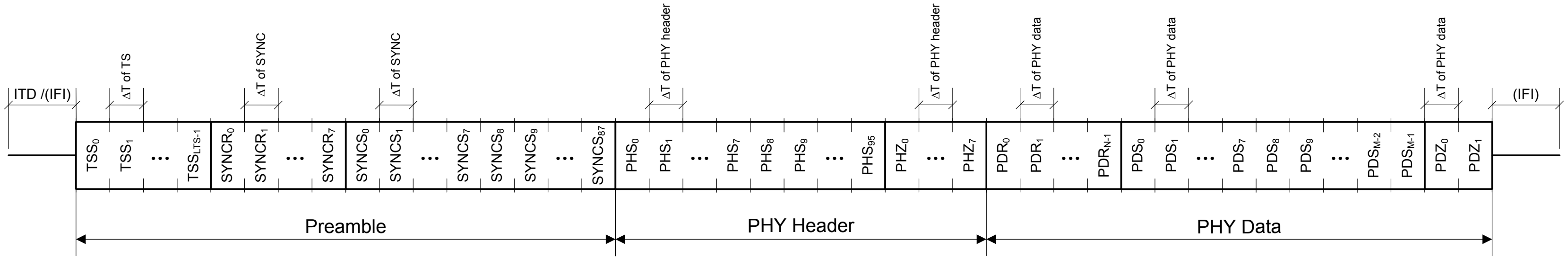

Figure 5 – Structure of transmit signal (PHY frame) consisting of overlapped modulated symbols

5.2.1.2 Preamble

Each PHY frame starts with the preamble, which is used for bit and block synchronization and for automatic gain control (AGC) adaptation. As shown in Figure 5, the preamble starts with a training sequence (TS) of symbols TSS_0 to TSS_{LTS-1}, followed by reference symbols $SYNCR_0$ to $SYNCR_7$ and symbols $SYNCS_0$ to $SYNCS_{87}$ of the synchronisation sequence.

The modulated TS symbols TSS_0 to TSS_{LTS-1} are generated using the corresponding parameters from Table 2 and the parameter TLS and TSA (see. Annex D), that are defining the PSK encoding of the TS. Note that neither FEC-encoding nor interleaving is performed on the TS (see Table 2).

The modulated symbols of the synchronisation sequence $SYNCS_i$, i = 0, …, 87 are generated by repetition encoding and differential modulation of SYNC, which is defined as the following bit sequence (cf. Table 1):

$$SYNC = (d_0, \ldots, d_{10}) = (1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0).$$

The symbols $SYNCR_0$ to $SYNCR_7$ serve as initial reference symbols for differential modulation of the first eight symbols $SYNCS_0$ to $SYNCS_7$ respectively. The initial carrier frequency phases of the modulated symbols $SYNCR_0$ to $SYNCR_7$ are defined by the parameter array RPA0 (see Annex D).

5.2.2 PHY header

The PHY header starts with the bits of the transmit mode (TM) field. These bits are denoted as d_{11}, d_{12}, …, d_{15} They indicate the number of the transmission mode, which is used to transmit data contained within the field PHY Data. The bits d_{11} to d_{22} (fields TM, Res. and HTB) of the PHY header are convolutionally encoded with the rate 1/8, which is defined in the sub-section 5.3.2.3. The purpose of the zero PHY header tail bits (HTB) d_{17} to d_{22}, is to return the encoder to its initial state. The encoded bit sequence is interleaved and mapped to the phase angels and then to the carrier frequencies according to the DBPSK rule and the frequency mapping scheme FMS1 respectively. The preamble symbols $SYNCS_{80}$ to $SYNCS_{87}$ are used as references for differential modulation of the first eight symbols PHS_0 to PHS_7 respectively.

At the end of the PHY header, eight zero symbols PHZ_0 to PHZ_7 are inserted in order to ensure that the modulated symbols of the PHY header and PHY data do not overlap in time. This is necessary because the PHY data may be transmitted in a different transmission mode than the PHY header. A zero symbol is a symbol which consists of all zero samples for the duration of T_S, where T_S is defined in Table 2 (cf. Figure 8).The transmit mode for the PHY data is defined by the field TM in the PHY header and can be any of transmit modes TM0 to TM20 from Table 2. The parameter of the transmit mode used for the PHY header are the same as of the PHY data transmit mode TM0 (cf. Table 2).

5.2.3 PHY data

The PHY-data field always contains a PHY service data unit (PSDU). Furthermore, for transmit modes TM0 to TM16, it also contains six zero tail bits (TB) in order to return the convolutional encoder to its initial state. For TM16 to TM20 there is no TB field because a repetition block code is used, see Table 2. It needs to be noted that the total size of PSDU can vary between 96 bits and 2152 bits.

As explained in the previous chapter, the transmit mode of the PHY data is defined by the TM field of the PHY header.

The bits d_{23} up to d_{L+28} (fields PSDU and, if available, TB) of the PHY data are FEC encoded. The encoded bit sequence is interleaved and mapped to the phase angels and then to the carrier frequencies according to the used DPSK rule and the frequency mapping scheme respectively.

The first N symbols of PDR_0 to PDR_{N-1} serve as initial reference symbols for differential modulation of the corresponding carrier of the first N symbols PDS_0 to PDS_{N-1} respectively. The number N is equal to the length of the frequency hopping cycle, which is the number of modulated symbols contained therein (Nfhc). It depends on the number of used carrier frequencies (Ncf) and the number of simultaneously used carriers (Nsuc) in the PHY data transmission mode. The value of Nfhc is defined as: Nfhc = Ncf / Nsuc = 1, 3, 5, 7 or 8 (cf. Table 2). The initial carrier frequency phases of the modulated symbols PDR_0 to PDR_0 to PDR_{N-1} , N = 1, 3, 5, 7 or 8 are user-specific.

At the end of the PHY data two additional zero symbolsPDZ_0 and PDZ_1 are transmitted (see zero symbol definition in Sub-section 5.2.2). This ensures a complete transmission of the previous symbols PDS_{M-2} and PDS_{M-1} (see Figure 8).

5.3 PHY frame transmission

5.3.1 General

The PHY layer of the AMC-SS allows the transmission of the PSDU using one of the 21 transmit modes TM0 to TM20. They differ from each other by the used FEC code, type of DPSK, frequency hopping scheme etc. (see Table 2 below), allowing a flexible trade-off between bit-rate and noise immunity. This is necessary for the adaptation of the MAC layer to the conditions of the communication channel.

The TS and SYNC of the preamble (PRMB) and the PHY header are transmitted in the predefined transmit modes. The transmit mode for the PHY data field is determined by the MAC entity using the primitive P_data.request (see Sub-section 5.5). The number of this mode is transmitted in the TM field of the PHY header to indicate it to the receiver for a correct reception of the PHY data.

The following Table 2 contains the parameters of the transmit modes for the preamble, PHY header and PHY data field. The transmit modes TM0 to TM15 are arranged by increasing data bit rate. The additionally defined transmission modes TM16 to TM20 use more simple transmission schemes allowing a low-cost implementation of the receiver.

The PHY layer processing steps during the transmission of the PHY frame as shown in Figure 3 are described in following sub-sections.

Table 2 – Parameter of preamble, PHY header and PHY data transmit modes

		Nominal data bit rate bit/s	Lower -3dB-frequency of TX-signal kHz	Upper -3dB-frequency of TX-signal kHz	Carrier frequency spacing kHz	Number of carrier frequencies	Number of simultaneously used carriers	Frequency hopping cycle length symbols	Frequency mapping scheme (see Annex A)	Symbol-to-symbol shift ΔT (samples)	Symbol duration T_S (samples)	Window function W (see Annex A)	Type of modulation	Forward error correction code	Code rate	Interleaving block length L_{INTL} (see also A.2) bits
PRMB	TS	4800	39.0	89.5	6.764	8	1	8	FMS0	72	154	W0	PSK	--	1	--
	SYNC	600	39.0	89.5	6.764	8	1	8	FMS1	72	154	W0	DBPSK	RC	1/8	--
PHY header		600	39.0	89.5	6.764	8	1	8	FMS1	72	154	W0	DBPSK	CONV	1/8	96
PHY data	TM0	600	39.0	89.5	6.764	8	1	8	FMS1	72	154	W0	DBPSK	CONV	1/8	96
	TM1	850	41.5	88.7	10.850	5	1	5	FMS2	58	128	W1	DBPSK	CONV	1/7	120
	TM2	1200	39.0	89.5	6.764	8	1	8	FMS1	72	154	W0	DQPSK	CONV	1/8	192
	TM3	1600	39.0	89.5	6.764	8	1	8	FMS1	72	154	W0	DQPSK	CONV	1/6	192
	TM4	2400	41.5	88.7	10.850	5	1	5	FMS2	58	128	W1	DQPSK	CONV	1/5	240
	TM5	3200	41.5	88.7	10.850	8	1	8	FMS1	72	154	W0	DQPSK	CONV	1/3	192
	TM6	4000	41.5	88.7	10.850	5	1	5	FMS2	58	128	W1	DQPSK	CONV	1/3	240
	TM7	4800	39.0	89.5	6.764	8	1	8	FMS1	72	154	W0	DQPSK	CONV	1/2	180
	TM8	6000	41.5	88.7	10.850	5	1	5	FMS2	58	128	W1	DQPSK	CONV	1/2	150
	TM9	8000	44.2	88.7	2.7125	16	2	8	FMS3	58	128	W1	DQPSK	CONV	1/3	224
	TM10	9000	41.5	88.7	10.850	5	1	5	FMS2	58	128	W1	D8PSK	CONV	1/2	225
	TM11	12000	44.2	88.7	2.7125	16	2	8	FMS3	58	128	W1	DQPSK	CONV	1/2	224
	TM12	18000	44.2	88.7	2.7125	16	2	8	FMS3	58	128	W1	D8PSK	CONV	1/2	336
	TM13	27000	41.5	88.7	5.425	9	3	3	FMS4	58	128	W1	D8PSK	CONV	1/2	486
	TM14	32000	46.9	88.7	5.425	8	8	1	FMS5	128	128	W2	D8PSK	CONV	1/2	480
	TM15	64000	33.3	88.7	2.7125	20	20	1	FMS6	163	179	W3	D8PSK	CONV	1/2	600
	TM16	600	39.0	89.5	6.764	8	1	8	FMS1	72	154	W0	DBPSK	RC	1/8	128
	TM17	900	39.6	90.0	7.773	7	1	7	FMS7	55	134	W4	DBPSK	RC	1/7	98
	TM18	1200	41.5	88.7	10.850	5	1	5	FMS2	58	128	W1	DBPSK	RC	1/5	50
	TM19	1800	39.6	90.0	7.773	7	1	7	FMS7	55	134	W4	DQPSK	RC	1/7	192
	TM20	3000	40.1	87.4	10.629	5	1	5	FMS8	46	98	W5	DQPSK	RC	1/5	100

5.3.2 Forward error correction encoding

5.3.2.1 General

Two types of FEC are used by AMC-SS based on repetition block codes (RC) and convolutional codes (CONV). The bits d_0 to d_{10} of the synchronisation sequence SYNC are encoded using the repetition code with code rate 1/8. The PHY header bits d_{11} to d_{22} are encoded by the rate 1/8 convolutional encoder with constraint length K = 7, see Table 3. Depending on the used transmission mode, the PHY data bits are encoded by convolutional encoder (TM0 to TM15) or repetition encoder (TM16 to TM20). If repetition encoder is used, the PHY data block does not contain tail bits.

5.3.2.2 Repetition encoding

The repetition encoder uses the repetition block code $(N,1,N)_2$ with a rate of 1/N, with N being the number of used carrier frequencies (Ncf). The encoder repeats every bit d_k of the SYNCS or PHY data N times creating the code word of encoded bits $C_k = c_{k,0}, ..., c_{k,n}, ..., c_{k,N-1}$, $n = 0..N-1$, where C_k is defined as:

$$C_k = \begin{cases} c_{k,n} = 0, n = 0,...N-1 & \left| d_k = 0 \right. \\ c_{k,n} = 1, n = 0,...N-1 & \left| d_k = 1 \right. \end{cases}$$

The resulting encoded bits are denoted as $c_i = c_{k,n}, \quad i = N \cdot k + n, \quad n = 0,1, \ldots, N-1$.

5.3.2.3 Convolutional encoding

The six convolutional encoders that are based on binary convolutional codes with constraint length K=7 and code rates 1/N = 1/8, 1/7, 1/6, 1/5, 1/3 and 1/2 that are used for different transmission modes (see Table 2). The general structure of the convolutional encoder is shown in Figure 6.

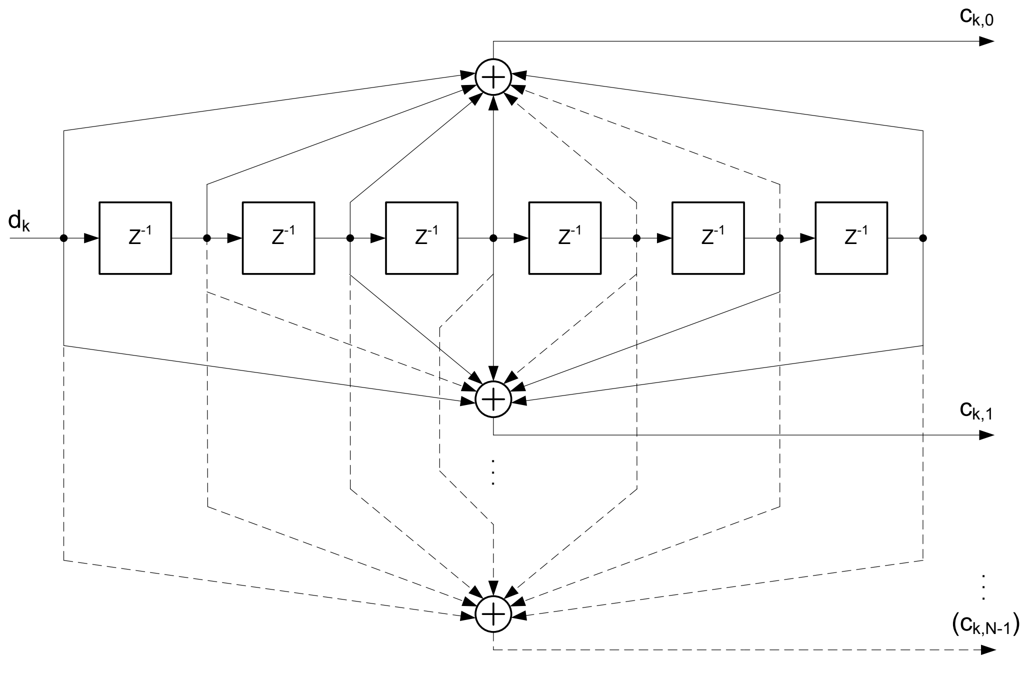

Figure 6 –General structure of a convolutional encoder with constraint length 7, used in this particular example (solid connections) for rate ½ encoder

Before the encoding of the PHY header, the initial state of the shift register in the encoder is zero. The code construction of each rate 1/N convolutional encoder is defined by the code connection vectors listed in Table 3.

Table 3 – Code connection vectors for convolutional encoder

		Code rate = 1/N					
		1/8	1/7	1/6	1/5	1/3	1/2
Code connection vectors (CCV_n) for output $c_{k,n}$, n = 0, 1, ..., N-1 of encoder	CCV_0	1101011	1110101	1111011	1111101	1011011	1111001
	CCV_1	1001001	1100111	1101001	1011001	1100101	1011011
	CCV_2	1110101	1111011	1011101	1011101	1111101	--
	CCV_3	1111011	1011101	1011101	1011101	--	--
	CCV_4	1011101	1011101	1110011	1100111	--	--
	CCV_4	1011101	1100111	1011111	--	--	--
	CCV_6	1100111	1011111	--	--	--	--
	CCV_7	1011111	--	--	--	--	--

Depending on the code connection vectors, the shift register taps are connected to N modulo2-summer generating the code word $C_k = (c_{k,0},\ ...,c_{k,n},\ ...,\ c_{k,N-1})$, $n = 0,1,\ \ldots,\ N-1$ from the bit d_k at the input of the encoder and bits d_{k-1} to d_{k-6} in the shit register stages. The solid connections in the encoder structure in Figure 6 correspond to the rate ½ encoder defined in Table 3. The dashed connections symbolise possible code connections in the encoder for other code rates.

The code words from the output of the encoder are serialised into a stream of encoded bits $c_i = c_{k,n},\ \ i = N \cdot k + n,\ \ \ n = 0,1,\ \ldots,\ N-1$.

With the shift of the last header tail bit d_{22} or the last tail bit d_{L+28} into the register, the encoder returns to its initial state finishing the encoding of the PHY header ore PHY data respectively (i.e., c_{183} is the last encoded bit of the PHY header). For encoding of the PHY data, the convolutional encoder shall be reinitialised according to a used transmit mode and with all zero in the shift register.

5.3.3 Interleaving

The encoded bits of PHY header and PHY data are segmented into input blocks of the length L_{INTL} $(sc_0,...,sc_i,...,sc_{L_{INTL}-1})$, where $sc_0 = c_i,...,sc_{L_{INTL}-1} = c_{i+L_{INTL}-1}$. Padding bits shall be added, if needed, to fill up the final block. Padding bits are generated by a pseudo-noise sequence generator shown in Figure 12 starting with all ones as initial state.

In the input block an intra-block interleaving is performed. The result is an output block $(oc_0,...,oc_p,...,oc_{L_{INTL}-1})$ containing the interleaved bits. Table 2 and Annex A.2 define the values of L_{INTL} and the corresponding bit interleaving schemes for the encoded PHY header and for the encoded PHY data in each transmit mode. Each cell of an interleaving scheme (i. e. interleaving table) defines the index i of the encoded bit sc_i from the input block $(sc_0,...,sc_i,...,sc_{L_{INTL}-1})$, which is put as the interleaved bit oc_p in the output block $(oc_0,...,oc_p,...,oc_{L_{INTL}-1})$. The cells of the interleaving scheme that are linked with the interleaved bits in the output block $(oc_0,...,oc_p,...,oc_{L_{INTL}-1})$, are sequentially arranged in rows starting in the 1st row from left to right and continuing from the left position in the next row. E.g., the 1st cell in the 1st row of an interleaving table is linked with the interleaved bit oc_0 in the output block. The interleaved bits are serialized as the bit sequence $(ic_0, ic_1, ..., ic_i, ...)$.

The bits of the encoded synchronisation sequence are not interleaved, i. e.: $(c_0, ..., c_{87}) = (ic_0, ..., ic_{87})$.

The encoded PHY header $c_{88}, ..., c_{183}$ is considered as one 96-bit-block interleaved according to the bit interleaving scheme defined in Annex A.2.

The interleaving schemes in Annex A.2 also show how the interleaved bits are mapped to the carrier frequencies.

5.3.4 PSK / DPSK mapping

The elements of the training sequence $(ts_0, ts_1, ..., ts_{LTS-1})$ and its length LTS are defined by the parameter array TSA and parameter TSL (see Annex D). An element of the TS is a nibble, which is defined as $(tsb3,\ tsb2,\ tsb1,\ tsb0)_i,\ i = 0,...,LTS-1$. It contains an encoded phase value for PSK. Using the phase mapping scheme for PSK from Table 4, the sequence $(ts_0, ts_1, ..., ts_{LTS-1})$ is transformed into the PSK-encoded training sequence $(ts\varphi_0, ts\varphi_1, ..., ts\varphi_{LTS-1})$.

After the PSK-encoded training sequence the reference-phase sequence $(\varphi_0, \varphi_1, ..., \varphi_7)$ is inserted. This sequence contains the use-defined values φ, $\varphi \in \{0, \pi/8, \pi/4, 3\pi/8, \pi/2, 5\pi/8, 3\pi/4, 7\pi/8, \pi, 9\pi/8, 5\pi/4, 11\pi/8, 3\pi/2, 13\pi/8, 7\pi/4, 15\pi/8\}$. It is necessary for DBPSK of the bit sequence $(ic_0, ic_1, ..., ic_i, ..., ic_{183})$ containing the encoded SYNC and PHY header, which is transmitted using eight carrier frequencies (see Table 2). Each bit of this bit sequence is considered in the DPSK mapper as the bit b0 in the Table 5 and mapped first to the corresponding value of $\Delta\varphi$, (i. e.: $ic_i,\ i = 0,...,183$ is mapped to $\Delta\varphi_i$). Then the DPSK mapper generates the PSK encoded sequence $(\varphi_0, \varphi_1, ..., \varphi_{i+8}, ..., \varphi_{191})$ by extending the reference-phase sequence $(\varphi_0, \varphi_1, ..., \varphi_i, ..., \varphi_7)$ with PSK elements $\varphi_8, \varphi_9, ..., \varphi_{i+8}, ..., \varphi_{191}$ that are calculated as $\varphi_i = \varphi_{i-8} + \Delta\varphi_i,\ i = 0,...,183$.

At the end of this sequence eight additional PSK elements $(\varphi_{192}, \varphi_{193}, ..., \varphi_{199})$ with the phase value $\pi/2$ are added to generate zero symbols PHZ_0 to PHZ_7 in the modulator.

Table 4 – Phase mapping scheme for PSK

tsb_3	tsb_2	tsb_1	tsb_0	φ
0	0	0	0	0
0	0	0	1	π/8
0	0	1	0	π/4
0	0	1	1	3π/8
0	1	0	0	π/2
0	1	0	1	5π/8
0	1	1	0	3π/4
0	1	1	1	7π/8
1	0	0	0	π/
1	0	0	1	9π/8
1	0	1	0	5π/4
1	0	1	1	11π/8
1	1	0	0	3π/2
1	1	0	1	13π/8
1	1	1	0	7π/4
1	1	1	1	15π/8

Then the DPSK mapper is reinitialised with the parameter of the transmit mode, which is used for the transmission of the remaining bit sequence $(ic_{184}, ic_{185}, ..., ic_i, ...)$ containing the encoded PHY data. Depending on the used transmit mode (DBPSK, DQPSK or D8PSK) the elements of this sequence are considered as single bits $ic_n = ic_i$ or grouped to bit duplets $(ic_i, ic_{i+1})_n$ or bit triplets $(ic_i, ic_{i+1}, ic_{i+2})_n$ respectively, where *n* is a consecutive number starting with 184.

In case of DBPSK, each single bit of this bit sequence is considered in the DPSK mapper as the bit b0 in the Table 5 and mapped first to the corresponding value of $\Delta\varphi$.

If DQPSK or D8PSK is used, each grouped bit duplet $(ic_i, ic_{i+1})_n$ or bit triplet $(ic_i, ic_{i+1}, ic_{i+2})_n$ of this bit sequence is considered in the DPSK mapper as the bit duplet (b0,b1) in

Table 6 or as the bit triplet (b0,b1,b2) in Table 7 respectively and mapped first to the corresponding value of $\Delta\varphi$.

Because the bit sequence $(ic_{184}, ic_{185}, ..., ic_i, ...)$ is transmitted without referencing to the PSK encoded sequence $(\varphi_0, \varphi_1, ..., \varphi_{i+8}, ..., \varphi_{199})$, an additional reference-phase sequence $(\varphi_{200}, \varphi_{201}, ..., \varphi_{\mathrm{Ncf}+199})$ is inserted, where Ncf is the number of carrier frequencies in the used transmit mode. This sequence contains values of φ, $\varphi \in \left\{0, \pi/8, \pi/4, 3\pi/8, \pi/2, 5\pi/8, 3\pi/4, 7\pi/8, \pi, 9\pi/8, 5\pi/4, 11\pi/8, 3\pi/2, 13\pi/8, 7\pi/4, 15\pi/8\right\}$ that are user-defined. Each of this phase values serves as reference for DPSK of the corresponding N_{FC} carrier frequencies in the used transmit mode.

As the next step the DPSK mapper generates the PSK encoded sequence $(\varphi_{200}, \varphi_{201}, ..., \varphi_{\text{Ncf}+199}, \varphi_{\text{Ncf}+200}, ..., \varphi_n, ...)$ by extending the reference-phase sequence $(\varphi_{200}, \varphi_{201}, ..., \varphi_{\text{Ncf}+199})$ with PSK elements of the sequence $(\varphi_{\text{Ncf}+200}, \varphi_{\text{Ncf}+201}, ..., \varphi_n, ...)$ that are calculated as $\varphi_n = \varphi_{n-\text{Ncf}} + \Delta\varphi_n$.

Table 5 – Phase difference mapping scheme for DPSK

b0	$\Delta\varphi$
0	0
1	π

Table 6 – Phase difference mapping scheme for DQPSK

b1	b0	$\Delta\varphi$
0	0	5π/4
0	1	7π/4
1	0	3π/4
1	1	π/4

Table 7 – Phase difference mapping scheme for D8PSK

b2	b1	b0	$\Delta\varphi$
0	0	0	11π/8
0	0	1	9π/8
0	1	0	13π/8
0	1	1	15π/8
1	0	0	5π/8
1	0	1	7π/8
1	1	0	3π/8
1	1	1	π/8

5.3.5 Carrier frequency mapping

During the transmission of the PPDU four frequency mapping schemes (FMS) are applied: FMS0 to the training sequence, FMS1 to the synchronisation sequence and PHY header, FMS1 to FMS8 to PHY data depending on the used transmit mode (see Table 2) and zero carrier frequency values to the zero symbols PHZ_0 to PHZ_7, PDZ_0 and PDZ_1. In the carrier frequency mapper the carrier frequency mapping the PSK-encoded training sequence $(ts\varphi_0, ts\varphi_1, ..., ts\varphi_{LTS-1})$ and the subsequent PSK-encoded sequence $(\varphi_0, \varphi_1, ..., \varphi_n, ...)$ of the remaining part of the PPDU are converted to a sequence of the unmodulated symbols $(V_0, V_1, ..., V_m, ..., V_{TNS-1})$, *TNS* is the total number of symbols in the PHY frame. The value of TNS depends on the length of the PSDU and the transmit mode, which is used for its transmission.

Depending on the used frequency mapping scheme (see A.3) and carrier window function an unmodulated symbol V_m is defined as:

$$V_m = \{\text{Nsuc}, \text{W}, c\varphi_{m,1}, cf_{m,1}, c\varphi_{m,2}, cf_{m,2}, \ ..., c\varphi_{m,NSUC}, cf_{m,NSUC}\}_m ,$$

where: Nsuc = 1, 2, 3, 8 or 20 is the number of simultaneously used carrier frequencies in the symbol; $c\varphi_{m,1}, c\varphi_{m,2}, \ ..., c\varphi_{m,NSUC}$ are the phases of this carriers, that are shaped by the window function *W*, and have the carrier frequencies $cf_{m,1}, cf_{m,2}, \ ..., cf_{m,NSUC}$ respectively; *W* indicates to the modulator which window function respectively symbol duration *Ts* and symbol-to-symbol shift *ΔT* are used.

For the preamble and PHY header Ncf = 8, Nsuc = 1, Nfhc = 8 and W = W0 is used (cf. 5.2.3). The unmodulated symbols V_0 to V_{TSL-1} of the training sequence in the preamble are built by the cyclic assigning of the carrier frequency values according to the carrier frequency mapping scheme FMS0 to the PSK-encoded elements of the training sequence $(ts\varphi_0, ts\varphi_1, \ ..., ts\varphi_{LTS-1})$. The built sequence is denoted as $(V_0, \ ..., V_m, \ ..., V_{TNS-1})$,

where: $V_m = \{1, \text{W0}, c\varphi_{m,1}, cf_{m,1}\}_m$, $m = 0, ..., LTS-1$, $c\varphi_{0,1} = ts\varphi_0, \ ..., c\varphi_{LTS-1,1} = ts\varphi_{LTS-1}$, $cf_{0,1} = f_3, \ cf_{1,1} = f_6, \ ..., \ cf_{7,1} = f_8, \ cf_{8,1} = f_3, \ cf_{9,1} = f_6, \ ..., \ cf_{LTS-1,1} = f_8$.

Continuing the frequency mapping of the remaining part of the preamble and PHY header the carrier frequency values according to the carrier frequency mapping scheme FMS1 are cyclically assigned to the PSK-encoded sequence $(\varphi_0, \varphi_1, \ ..., \varphi_{191})$ and zero frequency values to the sequence $(\varphi_{192}, \varphi_{193}, \ ..., \varphi_{199})$. This is necessary to generate zero symbols PHZ_0 to PHZ_7 in the modulator. The continued sequence on the output of the carrier frequency mapper is denoted as: $(V_{LTS}, \ ..., V_m, \ ..., V_{LTS+199})$, where $V_m = \{1, \text{W0}, c\varphi_{m,1}, cf_{m,1}\}_m$, $m = LTS, ..., LTS+199$,
$c\varphi_{LTS,1} = \varphi_0, \ ..., c\varphi_{LTS+191,1} = \varphi_{191}, \ c\varphi_{LTS+192,1} = \pi/2, \ c\varphi_{LTS+193,1} = \pi/2, \ ..., c\varphi_{LTS+199,1} = \pi/2$,
$cf_{LTS,1} = f_1, \ cf_{LTS+1,1} = f_4, \ ..., \ cf_{LTS+7,1} = f_6, \ cf_{LTS+8,1} = f_1, \ cf_{LTS+9,1} = f_4, \ ..., \ cf_{LTS+191,1} = f_6$,
$cf_{LTS+192,1} = 0, \ cf_{LTS+193,1} = 0, \ ..., \ cf_{LTS+199,1} = 0$.

The carrier frequency mapping of the PHY data consist of the following three steps:

Step 1: Depending on the transmit mode carrier frequency $f_j \in \{f_1, f_2, \ ..., \ f_{\text{Ncf}}\}$ from the used FMS is cyclically, with the period Ncf, assigned to each element of the PSK-encoded sequence $(\varphi_{200}, \varphi_{201}, \ ..., \varphi_{\text{Ncf}+199}, \varphi_{\text{Ncf}+200}, \ ..., \varphi_n, \ ...)$. As result a new sequence of duplets $\{\varphi_n, f_j\}_n$, $n = 200, 201, \ ...$ is built.

If, e.g., TM13 with FMS4, Ncf = 9, Nsuc = 3, Nfhc = 3 and W = W0 is used, this new sequence may be denoted as:

$$(\{\varphi_{200}, f_1\}_{200}, \{\varphi_{201}, f_4\}_{201}, \{\varphi_{202}, f_7\}_{202}, \{\varphi_{203}, f_2\}_{203}, \{\varphi_{204}, f_5\}_{204}, \{\varphi_{205}, f_8\}_{205}, \ ...,$$
$$\{\varphi_{206}, f_3\}_{206}, \{\varphi_{207}, f_6\}_{207}, \{\varphi_{208}, f_9\}_{208}, \ \{\varphi_{209}, f_1\}_{209}, \ ..., \{\varphi_{217}, f_9\}_{217}, \{\varphi_{218}, f_1\}_{218}, \ ...).$$

Step 2: Within each carrier frequency cycle of Ncf duplets, the groups of Nsuc duplets are built and assigned cording to the used FMS to unmodulated symbols in the sequence $(V_{LTS+200}, V_{LTS+201}, ..., V_m, ..., V_{TNS-3})$. Therefore each carrier frequency cycle contains Nfhc = Ncf / Nsuc symbols.

In previous example the sequence on the output of the carrier frequency mapper would contain the unmodulated symbols
$V_m = \{3, \mathrm{W1}, c\varphi_{m,1}, cf_{m,1}, c\varphi_{m,2}, cf_{m,2}, c\varphi_{m,3}, cf_{m,3}\}_m$, where:
$m = LTS + 200,..., TNS(L, \mathrm{TM13}) - 3;$
$c\varphi_{LTS+200,1} = \varphi_{200}, c\varphi_{LTS+200,2} = \varphi_{201}, c\varphi_{LTS+200,3} = \varphi_{202},$
$c\varphi_{LTS+201,1} = \varphi_{203}, c\varphi_{LTS+201,2} = \varphi_{204}, c\varphi_{LTS+201,3} = \varphi_{205},$
$c\varphi_{LTS+202,1} = \varphi_{206}, c\varphi_{LTS+202,2} = \varphi_{207}, c\varphi_{LTS+202,3} = \varphi_{208},$
$c\varphi_{LTS+203,1} = \varphi_{209}, ..., c\varphi_{LTS+205,3} = \varphi_{217}, ...;$
$cf_{LTS+200,1} = f_1, cf_{LTS+200,2} = f_4, cf_{LTS+200,3} = f_7,$
$cf_{LTS+201,1} = f_2, cf_{LTS+201,2} = f_5, cf_{LTS+201,3} = f_8,$
$cf_{LTS+202,1} = f_3, cf_{LTS+202,2} = f_6, cf_{LTS+202,3} = f_9,$
$cf_{LTS+203,1} = f_1, ..., cf_{LTS+205,3} = f_9, ...;$
$TNS(L, TM13)$ is the number of unmodulated symbols in the PPDU transmitting a PSDU of L bit in the transmit mode TM13.

If, for example, TM14 with FMS5, Ncf = 8, Nsuc = 8, Nfhc = 1 and W = W3 is used for the transmission of the PHY data, then each carrier frequency cycle consist of only one symbol containing all eight carrier frequencies. It is denoted as:
$V_m = \{8, \mathrm{W3}, c\varphi_{m,1}, cf_{m,1}, c\varphi_{m,2}, cf_{m,2}, c\varphi_{m,3}, cf_{m,3}, c\varphi_{m,4}, cf_{m,4}, c\varphi_{m,5}, cf_{m,5}, c\varphi_{m,6}, cf_{m,6},$
$c\varphi_{m,7}, cf_{m,7}, c\varphi_{m,8}, cf_{m,8}\}_m$, $m = LTS + 200,..., TNS(L, \mathrm{TM14}) - 3;$

Step 3: As the last symbols V_{TNS-2} and V_{TNS-1} of the PHY data two unmodulated zero symbols are inserted. This ensures a complete transmission of the previous symbols in the modulator. They are denoted as:
$V_{TNS-2} = \{1, \mathrm{W}, \pi/2, 0\}_{TNS-2}$, $V_{TNS-2} = \{1, \mathrm{W}, \pi/2, 0\}_{TNS-1}$, where: *W* is a window function used in the transmit mode; *TNS* is the total number of symbols in the PHY frame.

5.3.6 Modulation

The transmit signal in the modulator is defined as a sequence of consecutive samples with the rate f_{Tx} = 347200 samples/s respectively the sampling interval $\Delta t_{Tx} = 1/ f_{Tx}$. The sampe rate may be an integer multiples of $f_{Tx}/2$, if it is required for the carrier frequency values in the parameter lists CFL0 to CFL8. In this case the window functions W0 to W5 shell be resampled. The over-all sample rate accuracy shall be better than ±25ppm.

Usually the PPDU transmission procedure starts with an initial transmission delay (ITD). It is defined by the parameter ITDD, which contains a value of N_{ITD} in multiples of the sampling interval Δt = 1/347200 s. During the ITD zero-samples $s(i) = 0$, i = 1 to N_{ITD} are transmitted.

In the modulator each unmodulated symbol V_m from the sequence $(V_0, V_1, ..., V_m, ..., V_{TNS-1})$ is first transformed to the corresponding modulated symbol

$S_m,\ m = 0,...,TNS-1$. The modulated symbols of the PHY frame shown in Figure 5 correspond to the symbols $S_m,\ m = 0,...,TNS-1$ as follows:

(S_0 to S_{LTS-1}) = (TSS_0 to TSS_{LTS-1}),
(S_{LTS} to S_{LTS+7}) = ($SYNCR_0$ to $SYNCR_7$),
(S_{LTS+8} to S_{LTS+95}) = ($SYNCS_0$ to $SYNCS_{87}$),
(S_{LTS+96} to $S_{LTS+191}$) = (PHS_0 to PHS_{95}),
($S_{LTS+192}$ to $S_{LTS+199}$) = (PHZ_0 to PHZ_7),
($S_{LTS+200}$ to $S_{LTS+Nfhc+199}$) = (PDR_0 to PDR_{Nfhc}),
($S_{LTS+Nfhc+200}$ to S_{TNS-3}) = (PDS_0 to PDS_{M-1}),
(S_{TNS-3}, S_{TNS-1}) = (PDZ_0,toPDZ_1).

Where: LTS is the length of the training sequence in symbols; Nfhc is the length of the frequency hopping cycle in the PHY data transmit mode; TNS is the total number of symbols in the frame.

A symbol S_m consists of the consecutive samples $s_m(i),\ i = 1, ..., Ts$. *Ts* is the duration of the modulated symbol respectively window function *W* in samples. Using the modulation parameter Nsuc, *W*, $cf_{m,1}, cf_{m,2}, ..., cf_{m,NSUC}$, $c\varphi_{m,1}, c\varphi_{m,2}, ..., c\varphi_{m,NSUC}$ contained in the corresponding unmodulated symbol V_m (cf. Table 2) the sampling values $s_m(i)$ are defined as:

$$s_m(i) = \frac{-1}{\text{Nsuc}} \cdot \sum_{j=1}^{\text{Nsuc}} \left[w(i) \cdot \cos(2\pi \cdot cf_{m,j} \frac{i-0.5}{Ts} + c\varphi_{m,j}) \right],\ i = 1, ..., Ts\ ,$$

where: Nsuc = 1, 2, 3, 8 or 20 is the number of simultaneously used carrier frequencies in the symbol $cf_{m,1}, cf_{m,2}, ..., cf_{m,NSUC}$ with the phases $c\varphi_{m,1}, c\varphi_{m,2}, ..., c\varphi_{m,NSUC}$; $w(i)$ are samples of the window *W; Ts* is the duration of the symbol corresponding to the used *W*.

The samples of the resulting transmit signal are generated by summing up the samples of consecutive modulated symbols $S_m,\ m = 0,...,TNS-1$, that are shifted by ΔT samples against each other. The first sample $s_0(1)$ of the preamble follows the last zero sample of ITD. Figure 7 shows symbolically how the overlapped modulated symbols are combined with each other. As shown in this Figure, up to 3 modulated symbols can overlap.

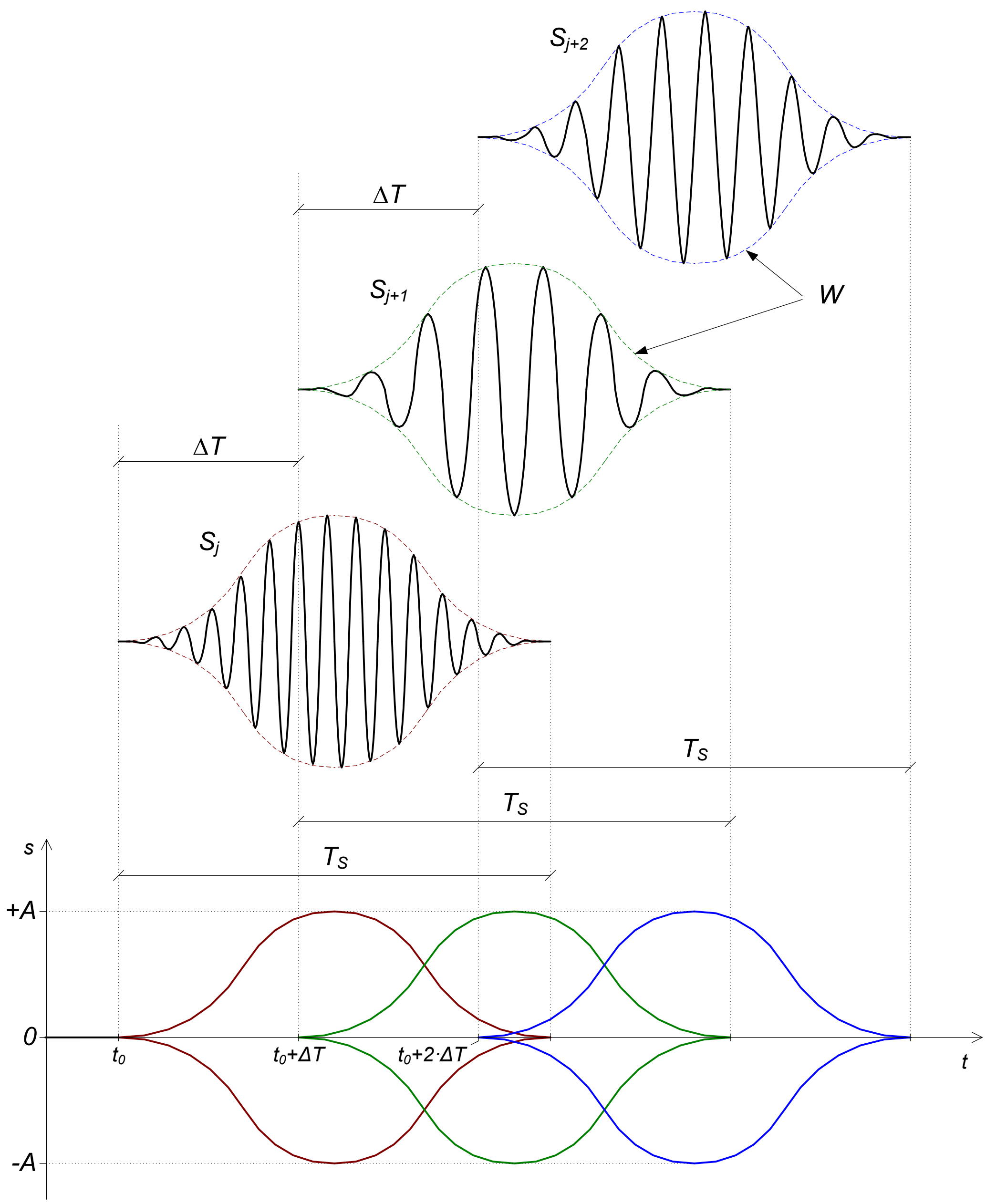

Figure 7 – Combining of overlapped modulated symbols

An interframe interval (IFI) of N_{IFI} sampling intervals Δt = 1/347200 s follows the last zero sample of the PDZ_1 symbol, if is requested by the MAC sub-layer entity. N_{IFI} is defined by the parameter IFID. During the IFI zero samples $s(i) = 0$, i = 1 to N_{ITD} are transmitted. In this case the next PHY frame shall be transmitted without ITD. Its first sample $s_0(1)$ of the preamble follows the last zero sample of IFI. Figure 8 shows symbolically how the last three PHY data symbols PDS_{M-3} to $PDS_{M-1}$of an PHY frame are combined with the zero symbols PDZ_0,toPDZ_1 , IFI and the first TSS_0 symbol of the next PHY frame.

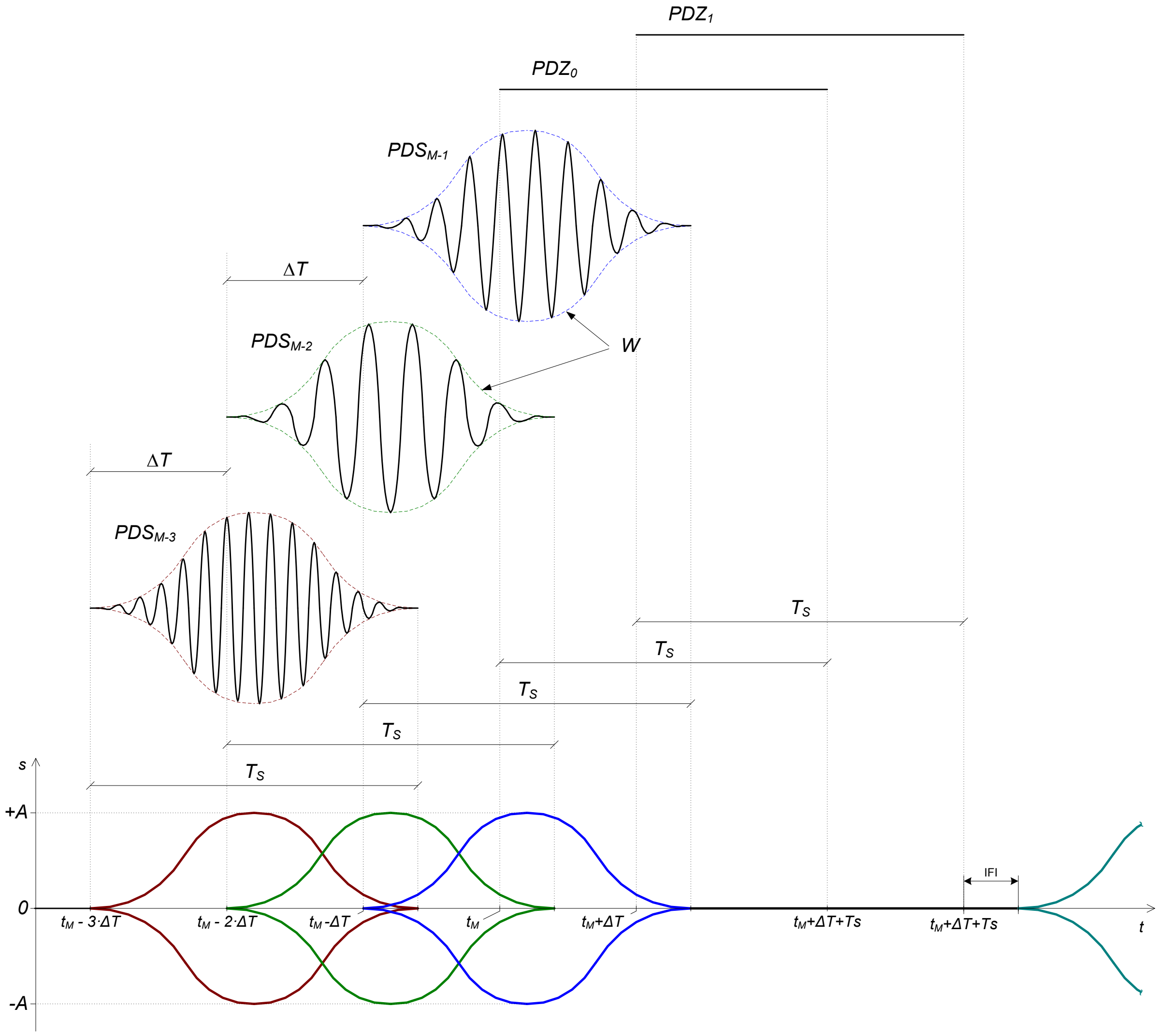

Figure 8 – Combining of overlapped modulated symbols followed by IFI

Finally the sampled transmit signal converted to the analogue signal and coupled into the physical channel – power-line.

5.4 EMC requirements

The emission requirements of EN 50065-1 shall be met.

5.5 PHY layer services

5.5.1 General

The PHY layer offers to the layer 2 only SEND/NO REPLY service.

As shown in Figure 9, the primitives P_data.request and P_data.indication are used to exchange information between the MAC sub-layer of layer 2 and the physical layer.

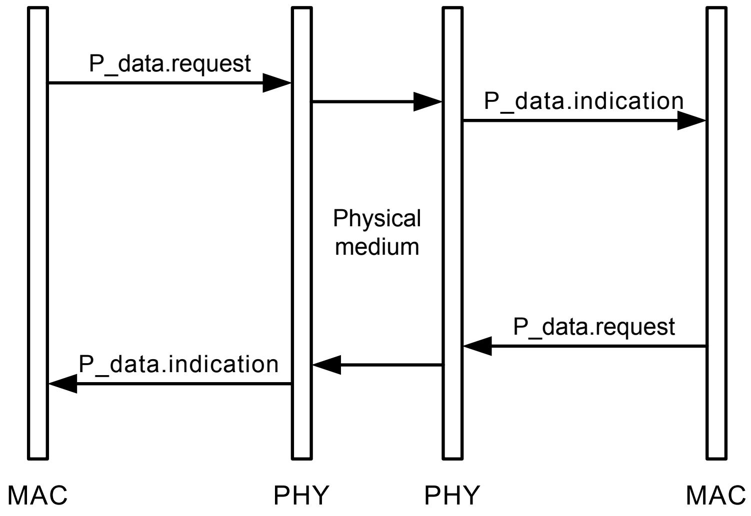

Figure 9 – Primitives between layer 2 and layer 1

The PHY layer primitive are described in the following sub-sections.

5.5.2 P_data.request

The P_data.request primitive is used by MAC sub-layer entity to initiate a PHY frame transmission procedure.

This primitive has following structure:

P_data.request{*TransmitTimingType, TransmitMode, DataLength, Data*}

The *TransmitTimingType* parameter specifies the type of the timing during the required PHY frame transmission procedure. The parameter may have one of the following values:

0: PHY frame shall be transmitted with ITD but without IFI;

1: PHY frame shall be transmitted with ITD and IFI at the end;

2: PHY frame shall start after the last zero sample of the last IFI. At the end of the frame the next IFI shall be transmitted;

3: PHY frame shall start after the last zero sample of the last IFI. At the end of the frame no IFI shall be transmitted.

The *TransmitMode* parameter specifies the transmit mode, which shall be used for the PSDU transmission in the PHY frame. The parameter may have one of the values 0 to 20 corresponding to the TM0 to TM20 respectively.

The *DataLength* parameter is the length of the *Data* parameter in bits. It may have any value in the range from 96 to 2152.

The *Data* parameter is a block of bits, which contains a MPDU to be transmitted.

5.5.3 P_data.indication

The P_data.indication primitive is used by a PHY layer entity to inform the MAC sub-layer entity about the reception of a PHY frame.

This primitive has following structure:

P_data.indication{*TransmitTimingType, TransmitMode, DataLength, Data*}

The *TransmitTimingType* parameter specifies the type of the PHY frame transmission timing. The parameter may have one of the following values:

0: PHY frame has been transmitted with ITD but without IFI;

1: PHY frame has been transmitted with ITD and IFI at the end;

2: PHY frame has been started after the last zero sample of the last IFI. At the end of the frame the IFI has been transmitted;

3: PHY frame has been started after the last zero sample of the last IFI. At the end of the frame no IFI has been transmitted.

The *TransmitMode* parameter specifies the transmit mode, which has been used for the PSDU transmission in the PHY frame. The parameter may have one of the values 0 to 20 corresponding to the TM0 to TM20 respectively.

The *DataLength* parameter is the length of the *Data* parameter in bits. It may have any value in the range from 96 to 2152.

The *Data* parameter is a block of bits, which contains the received PSDU = MPDU.

6 Data link layer specification

6.1 Overview

The MAC and LLC sub-layer of the data link layer provide point-to-multi-point and point-to-point communication on the low voltage distribution network between a master node and one or more slave nodes. The transmitted information is protected by CRC sequences. The MAC layer provides a simultaneous forwarding procedure in order to extend the achievable transmission range and to improve the transmission quality.

6.2 MAC protocol data unit

6.2.1 MPDU structure

An overview of the MPDU is shown in the Table 8.

Table 8 – Structure of MPDU

	Field name												
	Reserved bits	Frame forwarding sector number (FXS)	CIN of source node	NIN of source node	Link address (LA)	Data block length (DBL)	Total number of frame retransmission (FXT)	Frame retransmission down-counter (FXDC)	Reference-zero-crossing offset (RZCO)	LLC field (LLCF)	Frame header check sequence (FHCS)	Data block (DB)	Frame check sequence (FCS)
short frame (data block length = 0)	+	+	+	+	+	+	+	+	+	+	+	--	--
data frame (data block length ≠ 0)	+	+	+	+	+	+	+	+	+	+	+	+	+

A paragraph containing a requirement.

NOTE “+”: frame contains the field; “-”: frame does not contain the field

A detailed structure of the MPDU, as shown in Figure 10, is defined in Table 9 and described in the following sub-sections.

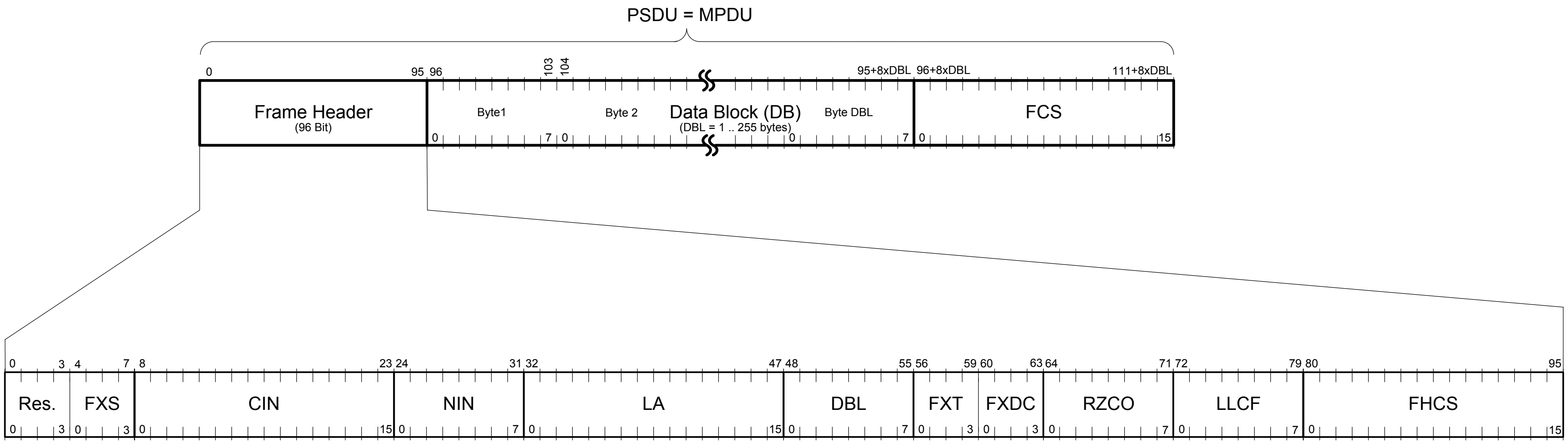

Figure 10 – MPDU structure

Table 9 – Description of MPDU

Field	Bit number	Bits	Full name of field	Value
Res.	3 to 0	4	Reserved bits for future use	0x0 - allowed value; 0x1 to 0xF - not allowed values.
FXS	7 to 4	4	Frame forwarding sector number	0x0 to 0xE – Fx-sector 0 to14; 0xF – frames shall be forwarded in all Fx-sectors.
CIN	23 to 8	16	MAC-channel identification number (CIN) of source node	0x0000 to 0xFFFE - allowed CIN values; 0xFFFF - broadcast to all network nodes with the same N_NIN.
NIN	31 to 24	8	Lower byte of the network identification number (NIN) of source node	0x00 to 0xFF - allowed values.
LA	47 to 32	16	Link address (LA)	0x0000 to 0xFFFD - valid link addresses; 0xFFFE - wait-state address; 0xFFFF - broadcast-address, which is used for broadcasts to all nodes with the same S_CIN within a network cell or, together with CIN = 0xFFFF for broadcast to all nodes with the same N_NIN within a network.
DBL	55 to 48	8	Data block length (DBL) in bytes	0x00 to 0xFF - allowed values.
FXT	59 to 56	4	Total number of frame retransmissions (FXT)	0x0 to 0xF - allowed values.
FXDC	63 to 60	4	Frame retransmission down-counter (FXCD)	0x0 to 0xF – allowed values if FXDC ≤ FXT.
RZCO	71 to 64	8	Reference-zero-crossing offset (RZCO)	0x00 to 0xFE - possible values; 0xFF – no RZCO available.
LLCF	79 to 72	8	Logical link control field (LLCF)	0x00 to 0xFF - possible values.
FHCS	95 to 80	16	Frame header check sequence	0x0000 to 0xFFFF - possible values.
DB	96 to 95+8xDBL	8xDBL	Data block (available if DBL ≠ 0)	0x00 to 0xFF - allowed values for each data block byte. The data block starts with the bit 0 of first byte and ends with the bit 7 of last byte.
FCS	111+8xDBL to 96+8xDBL	16	Frame check sequence (FCS) is based on CRC16 sequence., and is available if DBL ≠ 0	0x0000 to 0xFFFF - possible values.

6.2.2 Frame forwarding sector number

The field FXS contains the number of a frame forwarding sector. The frames with FXS ≠ 0xF are simultaneously forwarded only by nodes from this sector. The value 0xF is reserved for frame forwarding in all sectors simultaneously. The settings of the slave node contain a parameter S_FXS (Slave Frame Forwarding Sector) for its assignment to a frame forwarding sector. A group of the slave nodes, registered on a dedicated master node, with the same setting S_FXS ≠ 0xF, form a frame forwarding sector. Frames are forwarded/retransmitted only in one frame forwarding sector, if FXS = S_FXS ≠ 0xF. Frames with FXS = 0xF are forwarded by all slave nodes in a network cell independently from the value of S_FXS. The s-MN, alternatively acting as a slave, uses its own parameter S_FXS.

6.2.3 MAC-channel identification number

The field CIN contains the MAC-channel identification number. CIN≠0xFFFF identifies a dedicated master in the network. It is also used by slave nodes to address their master node. The CIN value 0xFFFF is reserved for broadcast from a master node to other master nodes within the same network. The settings of the master node contain a parameter ‘Master MAC-channel identification number (M_CIN)’, which is preconfigured with a unique 16 bit CIN≠0xFFFF. A Slave node contains a non-preconfigured dynamic parameter ‘Slave MAC-Channel identification number (S_CIN)’. During the registration (log-in) procedure the slave node receives from the master a new S_CIN. Registered slave nodes together with their master node create a network cell. S_CIN ≠ 0xFFFF is stored in a fail-safe memory as long as the slave node is registered with the master node. The current value of S_CIN is discarded (S_CIN=0xFFFF) after the slave’s registration with the current master has been lost or it becomes registered with another master with a new S_CIN. The s-MN can alternately act as master or slave using its own preconfigured M_CIN or an additional dynamic S_CIN respectively.

6.2.4 Network identification number

The field NIN contains the lower 8 bits of the 16 bit Network Identification Number of the Node (N_NIN) transmitting the frame. The same N_NIN value is preconfigured in all the nodes of the network.

6.2.5 Link address

The field LA contains the link address, which is used for layer 2 addressing. The value 0xFFFE is reserved for the wait-state address of a slave node. The value 0xFFFF is reserved for the broadcast address. With a LA ≠ 0xFFFF either one dedicated slave node or a group of slave nodes in a network cell (a so-called multicast group) is addressed by the master node. A slave node contains a non-preconfigured dynamic parameter ‘Slave link address (S_LA)’. Additionally the slave node may contain a non-preconfigured dynamic parameter array ‘Slave multicast addresses (S_MAx)’ with up to 31 slave multicast addresses: S_MA1 to S_MA31. The values of S_LA ≠ 0xFFFF and, if available, the values of S_MAx are assigned to the slave node by a master node during slave’s registration (log-in) procedure with the master. Note, that the value of S_LA shall be unique in a network cell. The (new) values of S_LA and S_MAx are stored in a fail-safe memory as long as the slave node is registered with the master node. The values of S_LA and S_MAx are discarded becoming = 0xFFFF after the slave’s registration with the current master has been lost. If the slave node becomes registered with another master with new S_LA and S_MAx values, the previous values of these parameters are discarded as well. A node acting as master does not have its own S_LA and S_MAx. An s-MN can alternatively act as slave node using its own S_LA and S_MAx assigned to it by the master node it is registered with.

6.2.6 Data block length

The field DBL determines the length of the data block in the frame in octets. If DBL = 0 the frame does not contain DB and FCS.

6.2.7 Total number of frame retransmissions

The field FXT provides the number of frame retransmissions in total. It is not changed when a frame is transmitted. If FXT = 0 the frame is transmitted only once without retransmissions.

6.2.8 Frame retransmission down counter

The field FXDC contains a value from 0 to 15. The FX down counter contains the number of remaining frame retransmissions. If FXDC ≠ 0, it is decremented before each frame retransmission by 1. A frame containing FXDC = 0, is transmitted only once without further retransmissions. For their initial transmission (before their retransmission), frames have to contain FXDC = FXT.

6.2.9 Reference zero-crossing delay

The field contains a value from 0 to 255. The Reference zero-crossing delay field contains a factor, which is multiplied with 34/347.2 ms to determine the duration from the last voltage zero-crossing on a predefined AC-line to the start of each frame transmission procedure. It may be analysed by slave or master nodes for AC-phase recognition. RZCO = 255 indicates that no reference zero-crossing delay is available.

6.2.10 Logical link control field

The field LLCF has two different formats, that depend on bit 7 of this field defining the frame transmission direction. Both formats of LLCF are shown in Figure 11.

7	6	5	4	3	2	1	0	
PRM =1	DLS	FCV	FCB	L_FC				Master → Master Master → Slave

7	6	5	4	3	2	1	0	
PRM =0	DLS	RES =0	DFC	L_FC				Slave → Master

Figure 11 – Formats of logical link control field

Elements of LLCF are defined in Table 10 to Table 14.

Table 10 – Description of LLCF (cf. IEC 60870-5-2)

LLCF element	Full name of element	Values
PRM	Primary bit	PRM=1 – message from master node; PRM=0 – message from slave node.
DLS	Data link service	DLS=0 - L_FC field contains function codes for data link control functions; DLS=1 - L_FC field contains function codes for higher layer servicing.
FCV	Frame count bit valid	FCV=1 - alternating function of the FCB bit is valid and can be analysed by slave node; FCV=0 - alternating function of the FCB bit is invalid. SEND/NO REPLY messages, MULTICAST or BROADCAST messages or other transmission services, that ignore the deletion of duplication or loss of information, do not alternate the FCB bit and indicate this by cleared FCV bit.
FCB	Frame count bit	0, 1 = alternating bit for successive SEND/CONFIRM or REQUEST/RESPOND messages directed to the same destination. The frame count bit is used to delete losses and duplication of information transfers. The master node alternates the FCB bit for each new SEND/CONFIRM or REQUEST/RESPOND message transmission directed to the same destination: to a slave node if DLS = 0 or to one of the link channels of the slave node if DLS = 1. Thus the master node keeps an own copy of the frame count bit for each of the destinations (each data link channel or data link control channel of a slave node). If an expected reply is timed out (missing) or garbled, then the same SEND/CONFIRM or REQUEST/RESPOND message is repeated with the same frame count bit. In case of 'Clear transmit buffer messages (L_FC = 0, DLS = 0,1) the FCB bit is always zero, and upon the receipt of these messages the data link control channel (DLS = 0) or one of link channels (DLS = 1, LCN = 0 to 15) of the addressed slave node will always be set to expect from the master node the next message, addressing the same channel, with FCV = 1 to have an opposite setting of FCB, it is FCB = 1.
RES	Reserved	RES=0 – valid value; RES=1 – invalid value
DFC	Data flow control	DFC=0 - further messages are acceptable; DFC=1 – indicates to the master node that an immediately following messages may cause buffer overflow in the slave node.
L_FC	Link function code	See Table 11 to Table 14 below for valid values

Table 11 – Link function codes if PRM=1 and DLS=0

Master node messages for data link control functions (PRM=1, DLS=0). See B.1.			
L_FC	**Type of message or service**	**Function**	**FCV**
0	SEND/NO REPLY expected	Generate temporary link address (BROADCAST)	0
1	SEND/CONFIRM expected	Clear transmit buffer	0
2	SEND/NO REPLY expected	Master node beacon (BROADCAST)	0
3	SEND/NO REPLY expected	Master node deactivated (BROADCAST)	0
4	SEND/NO REPLY expected	Clock synchronisation command (BROADCAST)	0
5	SEND/NO REPLY expected	Reuse link address (BROADCAST)	0
6	SEND/CONFIRM expected [a]	Perform PHY link test	1 / 0 [a]
7	SEND/NO REPLY expected	MN PHY link test	0
8	REQUEST/RESPOND expected	Request to node with temporary link address (BROADCAST)	0
9	REQUEST/RESPOND expected	Assign new address	1 / 0 [b]
10	REQUEST/RESPOND expected	Status enquiry	1
11	REQUEST/RESPOND expected	Node enquiry	1
12	REQUEST/RESPOND expected	Data availability enquiry (BROADCAST)	0
13, 14	-	Reserved for future use	-
15	REQUEST/RESPOND expected	Link quality enquiry	1

[a] The frame may be sent as a none-confirmed broadcast/multicast.
In this case no reply is expected and FCV bit is zero.

[b] The frame may be sent with LA=0xFFFF. In this case FCV bit is zero.

Table 12 – Link function codes if PRM=0 and DLS=0

Slave node messages for data link control functions (PRM=0, DLS=0). See B.3.		
L_FC	**Type of message or service**	**Function**
0	CONFIRM	ACK: Positive acknowledgement
1 to 6	-	Reserved for future use
7	RESPOND	SN PHY link test
8	RESPOND	Status response
9	RESPOND	Node response
10	RESPOND	Response from node with temporary link address
11	-	Reserved for future use
12	RESPOND	Data availability status
13, 14	-	Reserved for future use
15	RESPOND	Link quality response

Table 13 – Link function codes if PRM=1 and DLS=1

Master node messages for higher layer servicing (PRM=1, DLS=1). See B.2.			
L_FC	**Type of message or service**	**Function**	**FCV**
0	-	Reserved for future use	-
1	SEND-CONFIRM expected	Clear transmit buffer	0
2	-	Reserved for future use	-
3	SEND-CONFIRM expected	Data transmission	1
4	SEND-NO-REPLY expected	Data multicast (MULTICAST / BROADCAST)	0
5	SEND-CONFIRM expected	Segmented data transmission	1
6	SEND-NO-REPLY expected	Segmented data multicast (MULTICAST / BROADCAST)	0
7 to 10		Reserved for future use	-
11	REQUEST-RESPOND expected	Data enquiry	1
12 to 15	-	Reserved for future use	-

Table 14 – Link function codes if PRM=0 and DLS=1

Slave node messages for higher layer servicing (PRM=0, DLS=1). See B.4		
L_FC	**Type of message or service**	**Function**
0	CONFIRM	D_ACK: positive data reception acknowledgement
1 to 8	-	Reserved for future use
9	RESPOND	Data response
10 to 15	-	Reserved for future use

6.2.11 Frame header check sequence

The MAC header CRC sequence is calculated from the successive bits 11 to 16 of PPDU and bits 0 to 79 of MPDU and added at the end of the frame header. For calculation of a 16-bit CRC sequence, a polynomial $P_{CRC}(x) = x^{16} + x^{12} + x^5 + 1$ (also known as CRC CCiTT-16) is used, together with a 16-bit CRC initial value defined by parameter CRCINITVAL. The default value of CRCINITVAL = 0xFFFF.

6.2.12 Data block and frame check sequence

The Data block and the frame check sequence are available in the frame, if the data block length (DBL) is greater than zero. The frame check sequence is calculated from the data block bits, starting with the 1st bit of the 1st byte, as the second CRC sequence using the same polynomial as for the header CRC. The initial value for the frame check sequence calculation is the value of the frame header check sequence, meaning that CRC calculation just continues, with the header check sequence being an intermediate result. The frame check sequence is added immediately after the data block.

6.2.13 Scrambling

Before transmission, the MPDU is scrambled by a bit sequence SC with the length L_{SC}. MPDU and, if necessary, cyclic extended SC are bitwise combined using XOR operation starting with the 1st bit each. The bit sequence SC and its length are defined by parameter array SCA and the parameter SCL respectively.

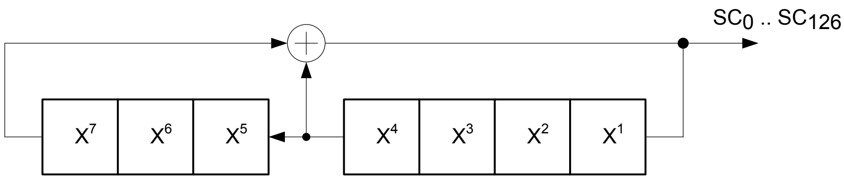

Figure 12 – Pseudo-noise sequence generator

Default values of SC are bits of a pseudo-noise sequence with the length L_{SC} = 127 generated at the output of a linear feedback shift register as shown in Figure 12, which is defined by the polynomial $P_{SC}(x) = x^7 + x^4 + 1$ and all ones as initial value of the register.

6.3 MAC frame transmission

The frames are retransmitted FXT times. The total number of retransmissions FXT, the transmission mode TM and, perhaps, the forwarding sector FXS are dynamically defined by the master node for each slave node. The retransmission improves the range and the frame transmission quality. Before the start of the frame transmission procedure by a master or a slave node, the FXT and FXDC in the frame shall have the same value. If this value is equal to zero, then the frame is sent only once. Otherwise the frame is retransmitted FXT times. Before every retransmission, the FXDC-value in the frame shall be decremented by 1 and the CRC-sequences FHCS and FCS shall be recalculated. An example of the frame transmission procedure of a SEND/NO REPLY-message is shown in Figure 13. In the example three copies of the same message have been received by the slave node. During the frame transmission procedure usually only the 1st received message copy is used. The others are discarded.

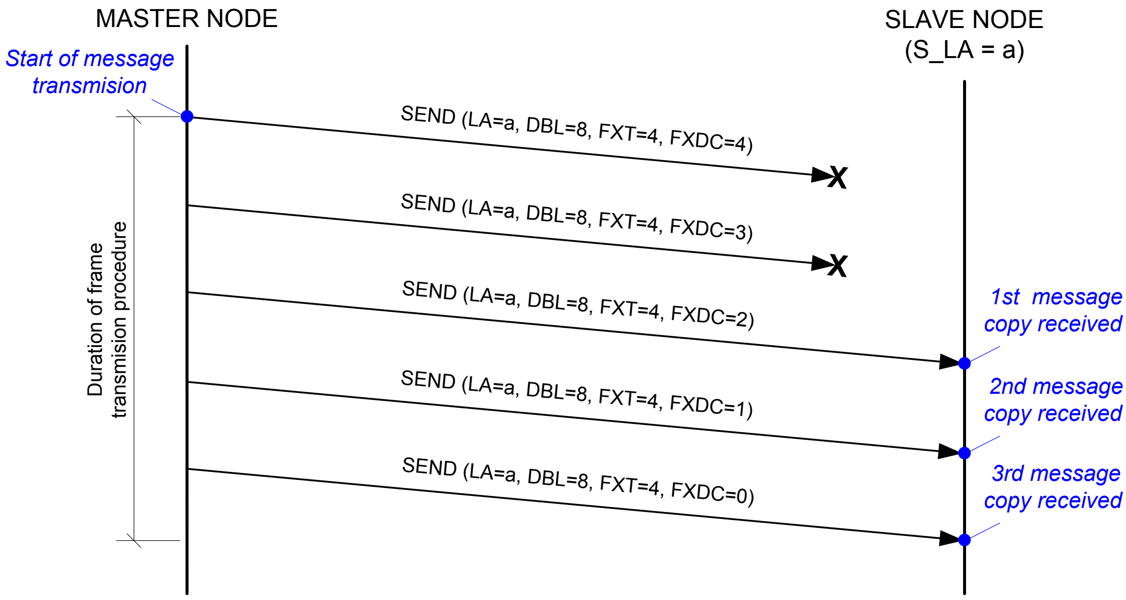

Figure 13 – Frame transmission procedure used by SEND/NO REPLY service.

Every frame transmission procedure in the MAC layer starts with an initial transmission delay (ITD) defined by the parameter ITDD. Between frame retransmissions lies an interframe interval (IFI). Its duration is defined by the parameter IFIT. Both parameters contain values in multiples of the sampling interval Δt = 1/347200 s.

For multi-hop transmission of the frames, the MAC layer provides a simultaneous forwarding procedure, where all involved nodes retransmit received frames simultaneously (synchronously). When a master node can reach a slave node without frame retransmission, this is called direct communication or communication in the hop plane 0. Each retransmission creates an additional hop plane. A slave node, which can be reached by the master node only via one slave node, is located on hop plane 1. Up to 16 hop planes (0 to 15 frame retransmissions) are possible.

Depending on the values of the parameters N_NIN, S_FXENA, S_FXS, S_CIN, S_LA, S_MAx, x = 1 to 31 and the values of FXS, CIN, NIN, LA, FXDC each network node acting as a slave node can forward a frame to other nodes. A network node acting as master node shall not forward frames from other network nodes. It retransmits a frame only during a transmission procedure initiated by itself.

If the value of the FX-decrement-counter in a MPDU received by a slave node is greater than zero, then the FX-down-counter shall be decremented by 1 and the frame shall be (re)transmitted until the FX-decrement-counter has reached a zero value. Only FXDC-values not greater than FXT are valid in received frames.

Transmission of frames with NIN ≠ L_NIN, L_NIN is lower byte of N_NIN value, is not allowed. A node discards frames with NIN ≠ L_NIN when received.

Usually nodes acting as slaves discard frames if S_CIN ≠ CIN ≠ 0xFFFF, but they may receive them, if it is necessary for the hand-off procedure.

A correctly received frame may only be forwarded by a slave node if at least one of the following conditions is true:

- (S_FXENA = 1) **AND** (FXDC ≠ 0x0) **AND** (CIN = 0xFFFF);

- (S_FXENA = 1) **AND** (FXDC ≠ 0x0) **AND** (S_CIN ≠ 0xFFFF) **AND** (S_LA ≠ 0xFFFF) **AND** (CIN ≠ 0xFFFF) **AND** (S_CIN = CIN) **AND** (LA= 0xFFFF);
- (S_FXENA = 1) **AND** (FXDC ≠ 0x0) **AND** (S_CIN ≠ 0xFFFF) **AND** (S_LA ≠ 0xFFFF) **AND** (CIN ≠ 0xFFFF) **AND** (S_CIN = CIN) **AND** (S_LA ≠ LA) **AND** (FXS = 0xF);
- (S_FXENA = 1) **AND** (FXDC ≠ 0x0) **AND** (S_CIN ≠ 0xFFFF) **AND** (S_LA ≠ 0xFFFF) **AND** (CIN ≠ 0xFFFF) **AND** (S_CIN = CIN) **AND** (S_LA ≠ LA) **AND** (S_FXS = FXS).

An example of simultaneous frame forwarding is shown in Figure 14. The transmission procedure was initiated by a master with a total number of frame retransmissions FXT=4 and FXDC=4. It means that the frame has been transmitted 5 times during this forwarding procedure: 4 frame retransmissions by other nodes follow the initial frame transmission from master node.

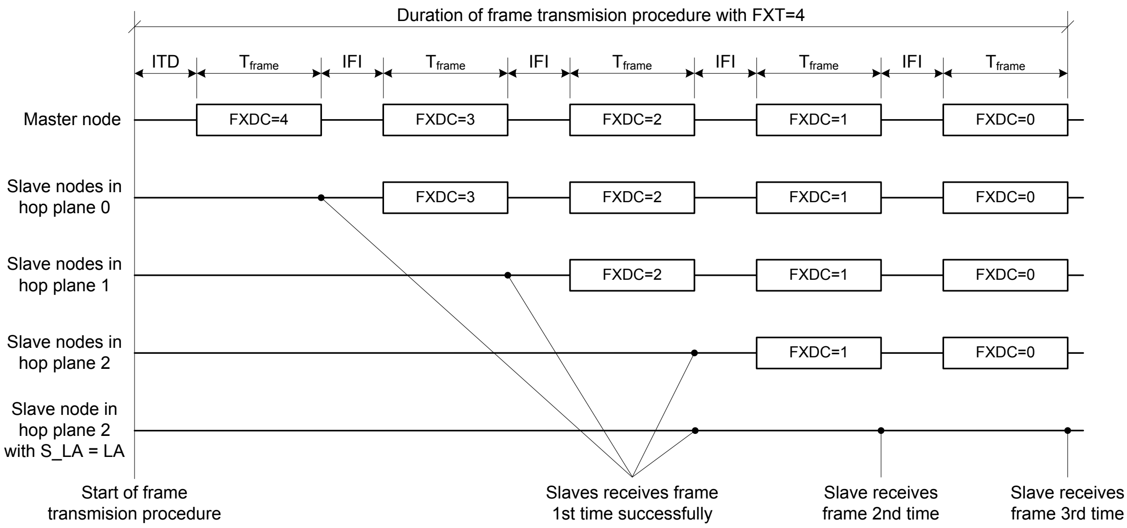

Figure 14 – Frame transmission procedure with simultaneous forwarding

For the first time the addressed slave node with S_LA = LA receives the frame already with FXDC=2. Two further frame receptions can follow until the end of the last frame retransmission with FXDC=0.

Only after the completion of the current frame transmission procedure, the slave node may answer by initiating a new frame transmission procedure to the master node.

Figure 15 shows an example of how the frame transmission procedure with simultaneous forwarding is used by REQUEST/RESPOND service in the LLC layer.

For the response the slave nodes use usually the same value of TM, FXS, CIN, NIN, LA and FXT as in the frame received from the master.

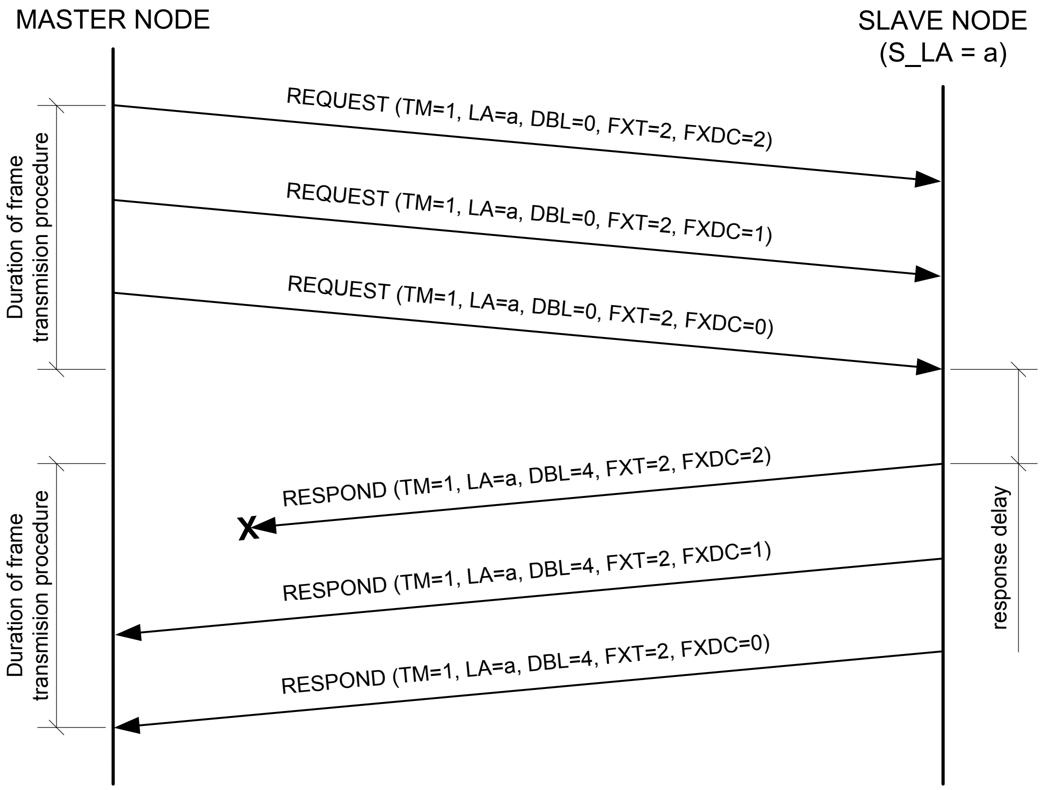

Figure 15 – Frame transmission procedures used by REQUEST/RESPOND service.

Transmission of the next REQUEST or SEND message by master node is only allowed after completion of all the expected frame retransmissions in the previous procedure.

6.4 The LLC protocol data unit

The LLC protocol data unit (LPDU) consists of the LLC-field transmitted in the frame header and the data block of the frame. The structure of LPDU, depending on the values in the LLC field, is defined in the normative Annex B.

6.5 Message transmission in LLC layer

6.5.1 General

For message transmission the LLC layer uses capability of the MAC layer to deliver frames from a slave node to a master node or from a master node to one or more slave nodes or from a master node to other master nodes. The master node controls the message traffic to and from the slave nodes.

The LLC-Layer offers two groups of services:

- LLC-services for control of the data link layer, which is structured as a cellular AMC-SS system, consisting of master nodes and associated slave nodes. For these services the data link control channel is used by setting the data link service bit (DLS = 0) to zero. The AMC-SS system is connected to the backbone network via the master nodes;
- LLC-services can be used by different higher layers entities for independent transmissions of their PDUs over the corresponding data link channels with

different priorities. These services are used with the set data link service bit (DLS = 1).

The LLC-services are described in the following sub-sections. The higher layer entities can access the services via the following primitives:

- DL_data.request;
- DL_data_identifier.confirm;
- DL_data.indication;
- DL_data.response;
- DL_data.confirm;
- DL_data_ack.response;
- DL_data_ack.confirm;
- DL_control.indication.

6.5.2 DL_data.request

The DL_data.request primitive is used by a higher layer entity to initiate a data transmission process.

This primitive has following structure:

DL_data.request{*ServiceType, CellIdentificationNumber, LinkAddress, LinkChannelNumber, DataPriority, DataLength, Data*}

The *ServiceType* parameter specifies the type of the required transmission. The parameter may have one of the following values:

0: non-acknowledged data transmission is requested. The value is only used by a master node;

1: acknowledged unicast data transmission with the *DataIdenifier* value 255 is requested;

2: acknowledged multicast/broadcast transmission with a *DataIdenifier* value other than 254 and 255, assigned by LLC is requested. Separate data acknowledgments over the LLC channel (DLS=0) are expected. The value is only used by a master node.

The *CellIdentificationNumber* parameter specifies the MAC-channel identification number of the transmission within a network cell. If this primitive is used by a master node, then the *CellIdentificationNumber* contains M_CIN of the master node. If this primitive is used by a slave node, then the *CellIdentificationNumber* contains the S_CIN value of the slave node. It may have any value in the range from 0x0000 to 0xFFFE.

The *LinkAddress* parameter specifies the link address for the transmission. If this primitive is used by a master node, then the *LinkAddress* parameter contains a link address of the destination: link address of a registered slave node (S_LA), one of the multicast addresses (S_MAx, x=1 to 31) or the broadcast address. If this primitive is used by a registered slave node, then the *LinkAddress* parameter contains the link address of the slave node (S_LA). It may have any value in the range from 0x0000 to 0xFFFF.

The *LinkChannelNumber* parameter specifies the link channel number for the transmission. It may have one of the following 16 values:

5: link channel number 5 is reserved for IPv4-PDUs;

6: link channel number 6 is reserved for IPv6-PDUs;

7: link channel number 7 is reserved for DLMS/COSEM-PDUs;

0 to 4 and 8 to 15: link channel number 0 to 4 and 8 to 15 are reserved for higher layer PDUs.

The *DataPriority* parameter indicates the priority of the data to be sent. It may have any of the following 16 values:

0: priority level 0 (lowest priority);

1 to 14: priority level 1 to 14 respectively;

15: priority level 15 (highest priority).

The *DataLength* parameter is the length of the *Data* parameter in octets. It depends on the value of parameter DBLMAX. For the *ServiceType* parameter values 0 and 1 the *DataLength* parameter may have any value in the range from 0 to 16 x (DBLMAX-3). For the *ServiceType* parameter value 2 it may have any value in the range from 0 to DBLMAX-2.

The *Data* parameter is a buffer of octets that contains the data to be transmitted. If the parameter *DataLength* is zero, the *Data* parameter does not contain any data.

6.5.3 DL_data_identifier.confirm

The DL_data_identifier.confirm primitive is used by a data link layer entity to confirm to the higher layer entity of the same node the assignment of a *DataIdentifier* value. These values correspond to the *ServiceType* value of the preceded DL_data.request.

This primitive has following structure:

DL_data_identifier.confirm {*Status, DataIdentifier, CellIdentificationNumber, LinkAddress, LinkChannelNumber, DataPriority, DataLength, Data* }

The *Status* parameter is used to pass the status information back to the requesting higher layer entity. It is used to indicate the success or failure of the associated previous DL_data.request primitive. The parameter may have one of the following values:

0: success of the associated previous DL_data.request;

1: failure of the previous associated DL_data.request. The transmit buffer of the requested link channel is occupied;

2: failure of the associated previous DL_data.request. There is no communication to the addressed node. The node registration is lost;

3: failure of the associated previous DL_data.request. The communication to the addressed node is temporarily lost.

The *DataIdentifier* parameter specifies an identifier of a data transmission service invocation. The *DataIdentifier* contains an identification number of the data transmission temporarily assigned by the LLC of the node to this data transmission service invocation. The *DataIdentifier* value may be reused for other service invocations only after the completion of the data transmission service or its abortive end (e.g. caused by a time-out). The parameter may have the following values:

0 to 199: number identifies multicast or broadcast transmissions 0 to 199 respectively, which is to be acknowledged by transmission of separate data acknowledgments over the data link control channel (DLS=0) from slave nodes during polling. The value is only used in messages from master nodes;

200 to 253: reserved for future use;

254: the number identifies a non-acknowledged transmission. The value is only used in messages from master nodes. There are no restrictions for its reuse;

255: the number identifies an acknowledged unicast transmission. There are no restrictions for its reuse.

The *CellIdentificationNumber* parameter specifies the MAC-channel identification number of the transmission within a network cell. If this primitive is used by a master node, then the *CellIdentificationNumber* contains M_CIN of the master node. If this primitive is used by a slave node, then the *CellIdentificationNumber* contains S_CIN value of the slave node. It may have any value in the range from 0x0000 to 0xFFFE.

The *LinkAddress* parameter specifies the link address for the transmission. If this primitive is used by a master node, then the *LinkAddress* parameter contains a link address of the destination: link address of a registered slave node (S_LA), one of the multicast addresses (S_MAx, x=1 to 31) or the broadcast address. If this primitive is used by a registered slave node, then the *LinkAddress* parameter contains the link address of the slave node (S_LA). It may have any value in the range from 0x0000 to 0xFFFF.

The *LinkChannelNumber* parameter specifies the link channel number for the transmission. It may have one of the following 16 values:

5: link channel number 5 is reserved for IPv4-PDUs;

6: link channel number 6 is reserved for IPv6-PDUs;

7: link channel number 7 is reserved for DLMS/COSEM-PDUs;

0 to 4 and 8 to 15: link channel number 0 to 4 and 8 to 15 are reserved for higher layer PDUs.

The *DataPriority* parameter indicates the priority of the data to be sent. It may have any of the following 16 values:

0: priority level 0 (lowest priority);

1 to 14: priority level 1 to 14 respectively;

15: priority level 15 (highest priority).

The *DataLength* parameter is the length of the *Data* parameter in octets. It depends on the value of parameter DBLMAX. For the *DataIdentifier* parameter values 254 and 255 the *DataLength* parameter may have any value in the range from 0 to 16 x (DBLMAX-3). For the *DataIdentifier* parameter valuec 0 to 199 it may have any value in the range from 0 to DBLMAX-2.

The *Data* parameter is a buffer of octets that contains the data to be transmitted. If the parameter *DataLength* is zero, the *Data* parameter does not contain any data.

6.5.4 DL_data.indication

The DL_data.indication primitive informs the addressed higher layer entity about the reception of data.

This primitive has following structure:

DL_data.indication{*DataIdentifier, CellIdentificationNumber, LinkAddress, LinkChannelNumber, DataPriority, DataLength, Data*}

The *DataIdentifier* parameter specifies an identifier of a data transmission service invocation. The *DataIdentifier* contains an identification number of data transmission temporarily assigned by the LLC of the node to this data transmission service invocation. The *DataIdentifier* value may be reused for other service invocations only after the completition of the data transmission service or its abortive end (e.g. caused by a time-out). The parameter may have the following values:

0 to 199: number identifies multicast or broadcast transmissions 0 to 199 respectively, which is to be acknowledged by transmission of separate data acknowledgment over the data link control channel (DLS=0) from slave nodes during polling. The value is only used in messages from master nodes;

200 to 253: reserved for future use;

254: the number identifies a non-acknowledged transmission. The value is only used in messages from master node. There are no restrictions for its reuse;

255: the number identifies an acknowledged unicast transmission. There are no restrictions for its reuse.

The *CellIdentificationNumber* parameter specifies the MAC-channel identification number of the transmission within a network cell. The *CellIdentificationNumber* contains the M_CIN or S_CIN value of the master or slave node, which transmitted the message. The parameter may have any value in the range from 0x0000 to 0xFFFE.

The *LinkAddress* parameter specifies the link address of the transmission. If this primitive was received by a slave node, then the *LinkAddress* parameter contains a link address of the destination: link address of the slave node themselves (S_LA), one of the multicast addresses (S_MAx, x=1 to 31) or the broadcast address. If this primitive was received by a master node, then the *LinkAddress* parameter contains the link address of the slave node (S_LA), which transmitted the message. It may have any value in the range from 0x0000 to 0xFFFF.

The *LinkChannelNumber* parameter specifies the link channel number of the transmission. It may have one of the following 16 values:

5: link channel number 5 is reserved for IPv4-PDUs;

6: link channel number 6 is reserved for IPv6-PDUs;

7: link channel number 7 is reserved for DLMS/COSEM-PDUs;

0 to 4 and 8 to 15: link channel number 0 to 4 and 8 to 15 are reserved for higher layer PDUs.

The *DataPriority* parameter indicates the priority of the data that was received. It may have any of the following 16 values:

0: priority level 0 (lowest priority);

1 to 14: priority level 1 to 14 respectively;

15: priority level 15 (highest priority).

The *DataLength* parameter is the length of the *Data* parameter in octets. It depends on the value of parameter DBLMAX. For the *ServiceType* parameter values 0 and 1 the *DataLength* parameter may have any value in the range from 0 to 16 x (DBLMAX-3). For the *ServiceType* parameter value 2 it may have any value in the range from 0 to DBLMAX-2.

The *Data* parameter is a buffer of octets that contains the data that was received. If the parameter *DataLength* is zero, the *Data* parameter does not contain any data.

6.5.5 DL_data.response

The DL_data.response primitive is used by the addressed higher layer entity of a node to respond to a previously received DL_data.indication.

This primitive has following structure:

DL_data.response{*CellIdentificationNumber, LinkAddress,*
LinkChannelNumber, DataPriority}

The *CellIdentificationNumber* parameter specifies the MAC-channel identification number for the transmission within a network cell. The *CellIdentificationNumber* contains the M_CIN or S_CIN value of the master or slave node, which shall transmit the message. It may have any value in the range from 0x0000 to 0xFFFE.

The *LinkAddress* parameter specifies the link address for the transmission. If this primitive is used by a registered slave node, then the *LinkAddress* parameter contains the link address of the slave node (S_LA) responding to the master node. If this primitive is used by a master node, then the *LinkAddress* parameter contains a link address of a registered slave node (S_LA) the master node is responding to. It may have any value in the range from 0x0000 to 0xFFFD.

The *LinkChannelNumber* parameter specifies the link channel number for the transmission. It may have one of the following 16 values:

5: link channel number 5 is reserved for IPv4-PDUs;

6: link channel number 6 is reserved for IPv6-PDUs;

7: link channel number 7 is reserved for DLMS/COSEM-PDUs;

0 to 4 and 8 to 15: link channel number 0 to 4 and 8 to 15 are reserved for higher layer PDUs.

The *DataPriority* parameter indicates the priority of the message to be sent. It may have any of the following 16 values:

0: priority level 0 (lowest priority);

1 to 14: priority level 1 to 14 respectively;

15: priority level 15 (highest priority).

6.5.6 DL_data.confirm

The DL_data.confirm primitive informs the higher layer entity of the requesting node about the reception of a response.

This primitive has following structure:

DL_data.confirm{*CellIdentificationNumber, LinkAddress, LinkChannelNumber, DataPriority*}

The *CellIdentificationNumber* parameter specifies the MAC-channel identification number of the transmission within a network cell. The *CellIdentificationNumber* contains the M_CIN or S_CIN value of the master or slave node respectively, which transmitted the message. The parameter may have any value in the range from 0x0000 to 0xFFFE.

The *LinkAddress* parameter specifies the link address of the transmission. If this primitive is used by a slave node, then the *LinkAddress* parameter contains the link address of the slave node (S_LA). If this primitive is used by a master node, then the *LinkAddress* parameter contains the link address of the registered slave node (S_LA), from which the master node has received the message. It may have any value in the range from 0x0000 to 0xFFFD.

The *LinkChannelNumber* parameter specifies the link channel number of the transmission. It may have one of the following 16 values:

5: link channel number 5 is reserved for IPv4-PDUs;

6: link channel number 6 is reserved for IPv6-PDUs;

7: link channel number 7 is reserved for DLMS/COSEM-PDUs;

0 to 4 and 8 to 15: link channel number 0 to 4 and 8 to 15 are reserved for higher layer PDUs.

The *DataPriority* parameter indicates the priority of the message that was received. It may have any of the following 16 values:

0: priority level 0 (lowest priority);

1 to 14: priority level 1 to 14 respectively;

15: priority level 15 (highest priority).

6.5.7 DL_data_ack.response

The DL_data_ack.response primitive is used by the higher layer of a slave node to acknowledge a successful reception of a multicast/broadcast transmission with the *DataIdentifier* value in the range from 0 to 199.

This primitive has following structure:

DL_data_ack.response{*DataIdentifier, CellIdentificationNumber, LinkAddress, Status*}

The *DataIdentifier* parameter specifies an identifier of a data transmission service invocation. The *DataIdentifier* contains the *DataIdentifier* value of a multicast/broadcast data transmission from the preceded DL_data.indication requesting acknowledgment from the corresponding higher layer of the slave node. The parameter may have any value in the range from 0 to 199.

The *CellIdentificationNumber* parameter specifies the MAC-channel identification number for the transmission within a network cell. The *CellIdentificationNumber* contains the S_CIN value of the slave node. It may have any value in the range from 0x0000 to 0xFFFE.

The *LinkAddress* parameter specifies the link address of the transmission. The *LinkAddress* parameter contains the link address of the slave node (S_LA). It may have any value in the range from 0x0000 to 0xFFFD.

The *Status* parameter specifies the type of the required acknowledgment. The parameter may have one of the following values:

0: negative acknowledgment: data with the indicated *DataIdentifier* has not been received yet by the higher layer of the slave node;

1: positive acknowledgment: data with the indicated *DataIdentifier* was successfully received by the higher layer of the slave node.

6.5.8 DL_data_ack.confirm

The DL_data_ack.confirm primitive is used to inform the higher layer of the master node about the successful reception of an acknowledgment from the slave node to a mulicast/broadcast data transmission with the *DataIdentifier* value in the range from 0 to 199.

This primitive has following structure:

DL_data_ack.confirm{*DataIdentifier, CellIdentificationNumber, LinkAddress, Status*}

The *DataIdentifier* parameter specifies an identifier of a data transmission service invocation. The *DataIdentifier* contains the *DataIdentifier* value of a multicast/broadcast data transmission from the preceded DL_data.indication requesting acknowledge from the corresponding higher layer entity of the slave node. The parameter may have any value in the range from 0 to 199.

The *CellIdentificationNumber* parameter specifies the MAC-channel identification number for the transmission within a network cell. The *CellIdentificationNumber* contains the S_CIN value of the slave node, which transmitted the acknowledgment. It may have any value in the range from 0x0000 to 0xFFFE.

The *LinkAddress* parameter specifies the link address of the transmission. The *LinkAddress* parameter contains the link address of the slave node (S_LA), which transmitted the acknowledgment. It may have any value in the range from 0x0000 to 0xFFFD.

The *Status* parameter specifies the type of the acknowledgment. The parameter may have one of the following values:

0: negative acknowledgment: data with the indicated *DataIdentifier* has not been received yet by the higher layer of the slave node;

1: positive acknowledgment: data with the indicated *DataIdentifier* was successfully received by the higher layer of the slave node.

6.5.9 DL_control.indication

The DL_control.indication primitive informs the higher layer about the connection status to or of a slave node.

This primitive has following structure:

DL_control.indication{*Status, DeviceIdentifier, CellIdentificationNumber, LinkAddress, LinkChannelMap, MulticastAddressTotalNumber, MulticastAdressArray*}

The *Status* parameter indicates to the master or to the slave node the status of the connection between the slave node with *DeviceIdentificationNumber* and the master node, it registered with. The parameter may have one of the following values:

0: the slave node is registered with the master;

1: the slave node is no longer registered with the master;

2: communication between the slave node and the master node is temporarily lost.

The *DeviceIdentifier* parameter specifies the unique number of the slave node allowing its unambiguous identification in one or more networks. The *DeviceIdentifier* is organized as byte array starting with a one byte format number followed by the 14 bytes multivendor identification number (MIN) according to DIN 43863-5:2012-04 (compare the use of the format number and MIN in the messages in Annex B). The format number value is zero, indicating that *DeviceIdentifier* is based on the MIN according to DIN 43863-5. The format number values 1 to 255 are reserved for future use.

The *CellIdentificationNumber* parameter specifies the MAC-channel identification number of the slave node with the *DeviceIdentifier*, which is or was registered with the master node. The *CellIdentificationNumber* contains the S_CIN value of the slave node, which has or had been assigned to it by the master node. The parameter may have any value in the range from 0x0000 to 0xFFFE.

The *LinkAddress* parameter specifies the link address of the slave node with the *DeviceIdentifier*, which is or was registered with the master node. The *LinkAddress* parameter contains a link address of the slave node (S_LA), which has or had been assigned to it by the master node. The parameter may have any value in the range from 0x0000 to 0xFFFD.

The *LinkChannelMap* parameter specifies which link channels are used in the slave node with the *DeviceIdentifier*, which is or was registered with the master node. The *LinkChannelMap* parameter is a bitmap of 16 bits (bit 0 to bit 15) indicating if the link channel number 0 to 15 respectively are used by the slave node. Each bit of the *LinkChannelMap* parameter may have either one of the following two values:

0: link channel number n, n = 0 to 15 is out of use;

1: link channel number n, n = 0 to 15 is used.

The *MulticastAddressTotalNumber* parameter specifies the number of the multicastaddresses contained in the *MulticastAdressArray*. It may have any value in the range from 0 to 255. If the parameter *MulticastAddressTotalNumber* is zero, the *MulticastAdressArray* does not contain any multicast addresses.

The parameter array *MulticastAddressArray* specifies multicast addresses of the slave node. The *MulticastAddressArray* (MA1, MA2, …, MA_n, n = 1 ot 255) contains the multicast address values S_MAx = (S_MA1, S_MA2, …, S_MA_n), n = 1 to 31 of the slave node, that has or had been assigned to the slave node by the master node. Each of the 16 bit multicast addresses may have any value in the range from 0x0000 to 0xFFFD.

6.5.10 Transmission from slave node

6.5.10.1 General

Data (PDUs of higher layer entities) transmission from slave nodes to a master node is controlled by the master node, using data acquisition services of the data link control channel (DLS = 0). The master node can acquire the slave's data or messages that are ready to be transmitted, using two methods: polling and quick-check procedure.

6.5.10.2 Data acquisition by polling

The master node periodically polls the registered slave nodes using the control channel of the LLC (DLS=0). Addressing of the registered slave nodes is based on LA, CIN and NIN in a message. Polling of the slave nodes by the master node is influenced by the following points:

- Number of registered slave nodes and number of used link channels.
  There is no polling if no slave nodes are registered.

- Data Priority (DP) of the frames.

- Sharing of transmission resources between data (DLS=1) and control (DLS=0) channels.

The data (PDUs) of higher layer entities (HLE) are stored in the data link layer (DLL) of the slave nodes and their availability is indicated to the master node when slave nodes are polled. As shown in Figure 16, first the master node sends the request 'Node enquiry' (DLS = 0, L_FC = 11). Then the addressed slave answers with the 'Node response' (DLS = 0, L_FC = 9) containing the following information:

- Number of link channel (LCN) containing data with higher priority ready for transmission and priority of data with corresponding LCN;

- Bit map with flags indicating data availability for one or more link channels identified by the link channel number;

- Bit map with flags confirming successful receptions of data transmitted before by multicast or broadcast transmission that require a response.

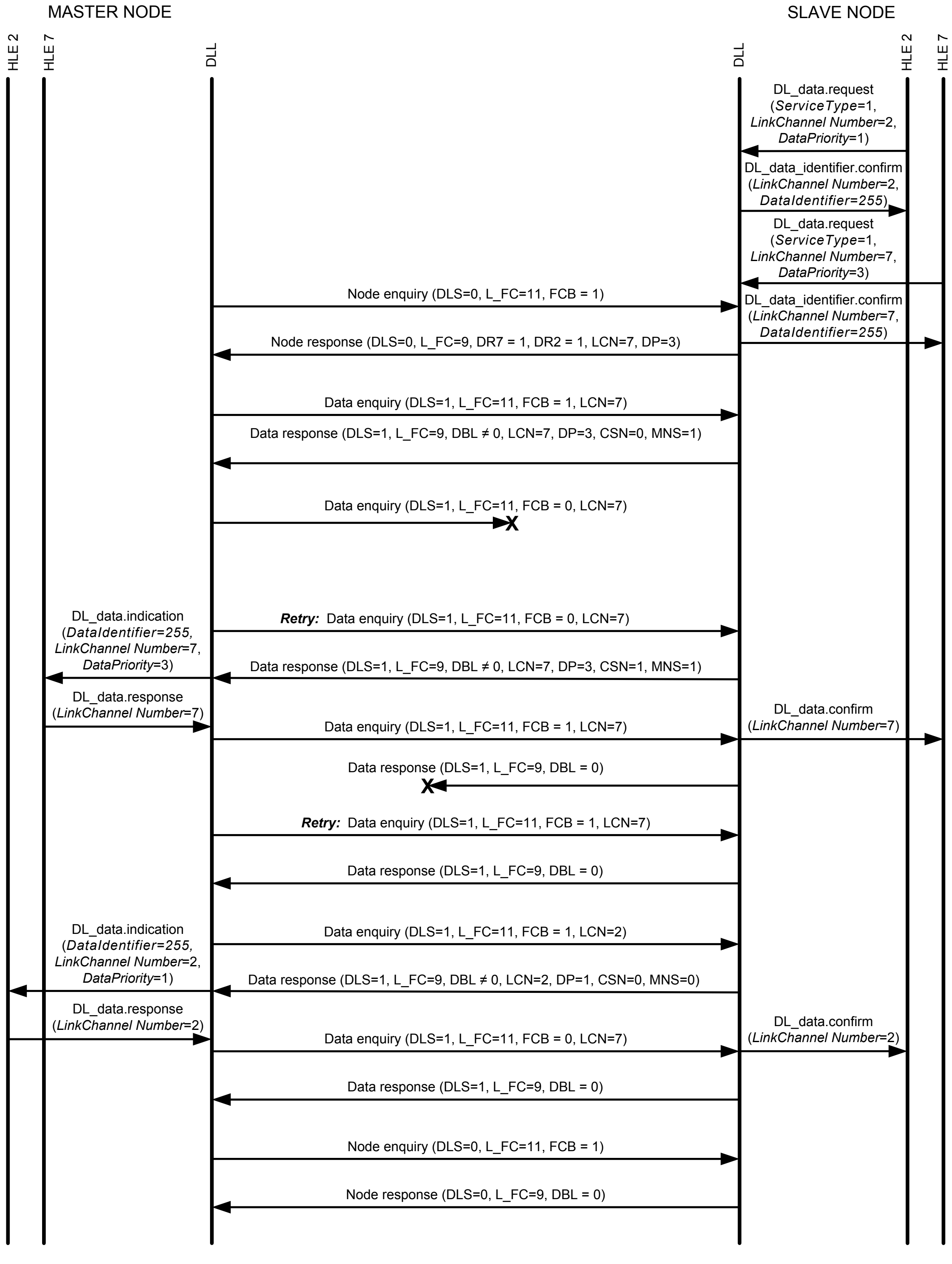

Figure 16 – Example of data collection by polling

If different higher layer entities of the slave node have data available (their DL_data.request are queued), the master node requests first transmission from the higher layer entity with the higher priority data by sending a ‘Data enquiry’ (DLS = 1, L_FC = 11). The addressed slave node’s entity answers with a ‘Data response’ (DLS = 1, L_FC = 9). The received response message is indicated to the corresponding higher layer entity of the master node with a DL_data.indication primitive. Immediately after the reception of this message, the master node entity confirms it using DL_data.response primitive. The response is transmitted to the slave node's entity by the next 'Data enquiry' (DLS = 1, L_FC = 11) from the master node with toggled FCB value. This is indicated to the corresponding slave node's entity using primitive DL_data.confirm.

If data from the slave node consist of more than one segment, several request-respond cycles are necessary to acquire this data. The slave node transmits the data segments in an ascending sequence of messages starting with the 1st message having CSN = 0. After a successful transmission of the last message in the sequence with CSN = MNS the transfer of the segmented data has been completed.

6.5.10.3 Data acquisition by quick-check procedure

This method can be used by a master node to accelerate the data acquisition time during the polling of slave nodes.

The master node broadcasts at periodic intervals 'Data availability enquiry' (L_FC=12) to all slave nodes registered in its cell (see Figure 17). These messages contain information about number and duration of one or more sequential contention windows as well as the information, if the transmission mode of the enquiry shall be reused by slave nodes for the response. Each slave node, which has data ready to transmit to the master node, randomly chooses one of the contention windows to respond with the message 'Data availability status' (L_FC=12). It also contains information about the link channel with higher priority data, its priority value and possibly a bitmap indicating data ready for transmission in other link channels.

By increasing the total number of contention windows (TNCW) and their duration (CWD), the master node can reduce the probability of collisions or garbled messages from slaves.

After the last contention window, the master node acquires data from those slave nodes, from which it has received indications. The higher priority data are acquired first. Data collection procedure is the same as described in Paragraph 6.5.10.2.

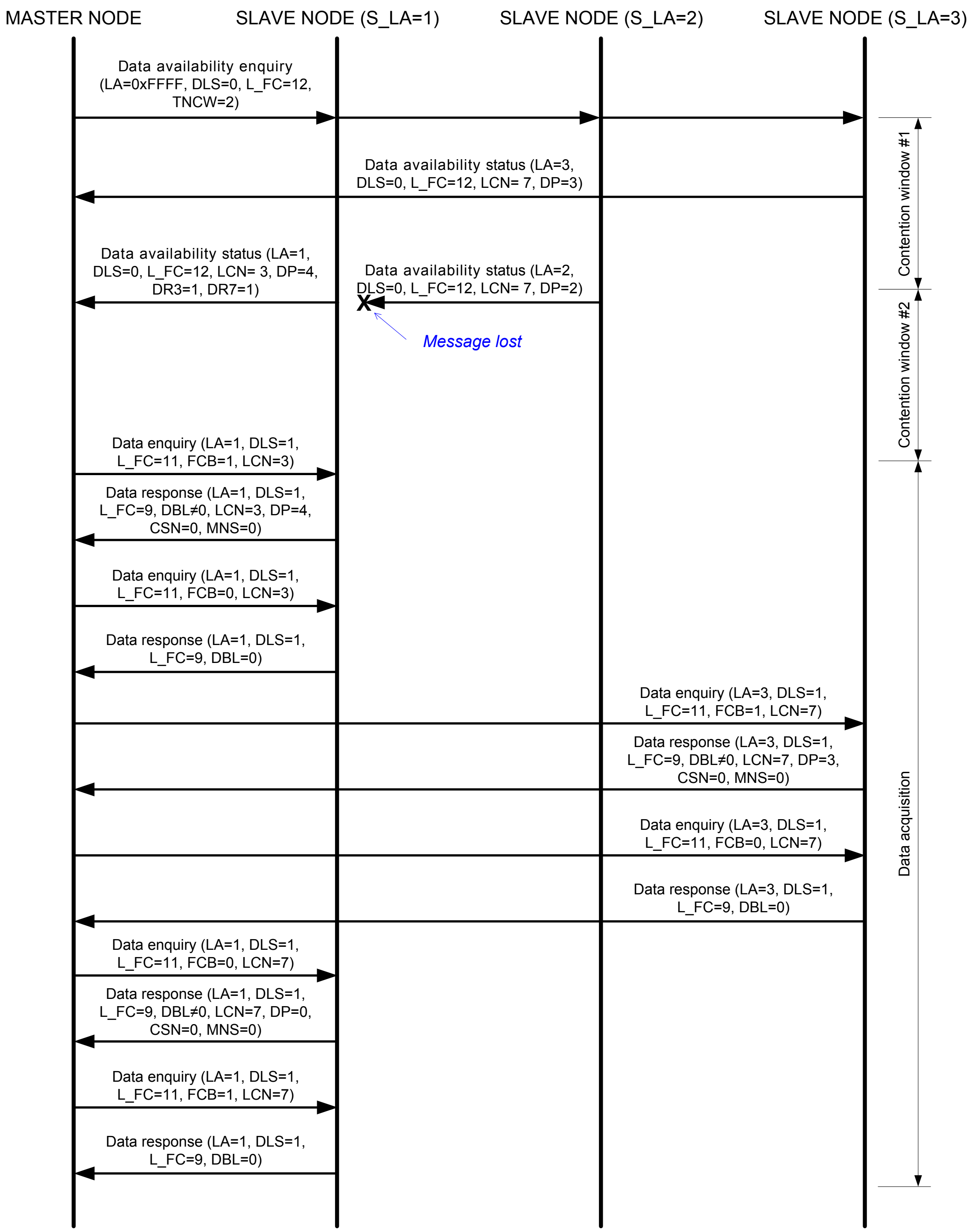

Figure 17 – Example of data collection using quick-check procedure

6.5.11 Transmission from master node

One or more DL_data.request containing data (PDUs) of higher layer entities (HLE) are queued in the data link layer (DLL) of the master node for transmission depending of its priority. According to the primitives described in detail before, a master node can send messages as:

- acknowledged unicast;

- acknowledged multicast or broadcast;
- non-acknowledged multicast or broadcast.

A multicast/broadcast transmission may only be used by nodes operating as master node. The distinction between unicast, multicast or broadcast is made by using either the link address of a dedicated slave node (S_LA) or one of multicast addresses (S_MAx, x=1 to 31) or a broadcast address according to the MAC address schema. During the registration of the slave node it is assigned to none, one or several multicast groups. Only 'Master node beacon' (L_FC = 2) message may be transmitted as broadcast to all nodes with the same N_NIN by using the control channel (DLS = 0) and the network wide broadcast address in the CIN-field in addition to the broadcast address in the LA-field.

The Multicast/Broadcast frames are usually transmitted with a bigger FXT value as in unicast frames. This FXT value is limited by the parameter FXTMAX.

For the multicast/broadcast transmission the master may possibly send several subsequent frames (repetitions) using different transmission modes (TM) in order to reach all addressed slave nodes with higher probability. Transmission modes with higher success rates are preferred. The master node tries to reach all the slave nodes with a minimum number of retries. The sequence of repetitions using different transmission modes is not interrupted by other DLL services.

As a slave node may receive the same multicast/broadcast frame with different TM and FXDC values, it shall have following capability to delete duplicated of messages:

- If the slave node is already registered with a master node, it accepts a received 'Data multicast' (DLS=1, L_FC=4, MCN=0 to 199) message only when its buffer, corresponding to the MCN value in the message, is free (i. e.: the previous message in the buffer was successfully acknowledged by the slave node);
- If the slave node is not registered with the master node, it accepts a received 'Data multicast' (DLS=1, L_FC=4, MCN=0 to 199) message only then, if its DB content is either different from the last accepted 'Data multicast' message with the same MCN value or without comparing of DB, if the message has been received beyond the time period T_{MCTMAX} , which is defined by the parameter MCTMAX, after the reception of this last accepted message;
- If the slave node is (not) registered with the master node, it accepts a received 'Data multicast' (DLS=1, L_FC=4, MCN=254) or 'Segmented data multicast' (DLS=1, L_FC=6, MCN=254) message only then, if it's DB content is either different from the last accepted 'Data multicast'/'Segmented data multicast' or without comparing of DB, if the message has been received beyond the time period T_{MCTMAX} , which is defined by the parameter MCTMAX, after the reception of this last accepted message.

The LLC-sub-layer offers two types of multicast/broadcast services: acknowledged and unacknowledged multicast/broadcast message transmission. Only master node messages 'Data multicast' (L_FC=4) containing a MCN value between 0 and 199 shall be acknowledged by addressed slave nodes.

A MCN value in the range from 0 to 199 and the corresponding transmit and receive buffers of master and slave node respectively may be reused, if the acknowledged multicast/broadcast transmission was successfully finished (including selective retransmissions of frames) or a acknowledged multicast/broadcast time-out period T_{AMCTO} , which is defined by the parameter AMCTO, has expired. Up to 200 multicast/broadcast transmission procedures can be active at the same time. The assignment of the multicast-buffer to different data link channels is implementation specific.

For an unacknowledged multicast/broadcast transmission the MCN value 254 is used. The MCN value 255 is used for a unicast transmission.

6.5.12 Acknowledged unicast transmission

6.5.12.1 General

A unicast message contains a (new) link address value of a dedicated slave node, which it is directed to. The slave's response/confirm message acknowledges a successful reception of the master's request/send message.

A received 'Data transmission' (DLS=1, L_FC=3, MCN=0 to 199) or 'Segmented data transmission' (DLS=1, L_FC=5, MCN=0 to 199) message is indicated to the addressed higher layer entity of the slave node with a DL_data.indication primitive (see Figure 18). Immediately after the reception of this message, the slave node entity confirms it using DL_data.response primitive. The response is transmitted to the master node using the 'D_ACK' (DLS=1, L_FC=0) message and finally indicated to the corresponding entity using primitive DL_data.confirm.

If data from the master node consists of more than one segment, several send-confirm cycles are necessary to transmit this data to the slave node. The master node transmits the data segments in an ascending sequence of messages starting with the 1st message having CSN = 0. After a successful transmission of the last message in the sequence with CSN = MNS the transfer of the segmented data is completed.

The maximal duration from the start of the unicast transmission (request/send message) to the reception of a slave node's answer (respond/confirm message) by the master node is calculated as follows (cf. Figure 15):

$$T_{UC\_ACK\_MAX} = T_{request\_duration} + T_{max\_response\_delay} + T_{max\_response\_duration} ,$$

where: $T_{request\_duration}$ is a duration of REQUEST or SEND message transmission procedure from the master node to the addressed slave node;

$T_{max\_response\_duration}$ is an estimated maximal duration of the corresponding RESPOND or CONFIRM message transmission procedure from the addressed slave node to the master node;

$T_{max\_response\_delay}$ is a maximal delay in the slave node to prepare the corresponding RESPOND or CONFIRM message for its transmission to the master node.

Respond/confirm message from the slave node to the master node are acknowledged by the master node in the next frame by toggled FCB bit (cf. Table 10, Figure 16 and Figure 18). Slave nodes evaluate the value of FCB bit, if the FCV bit is set to 1. In addition to the FCB bit for the control channel (DLS = 0) the master node administer independently separate FCB bits for each link-channel identified by LCN (DLS = 1).

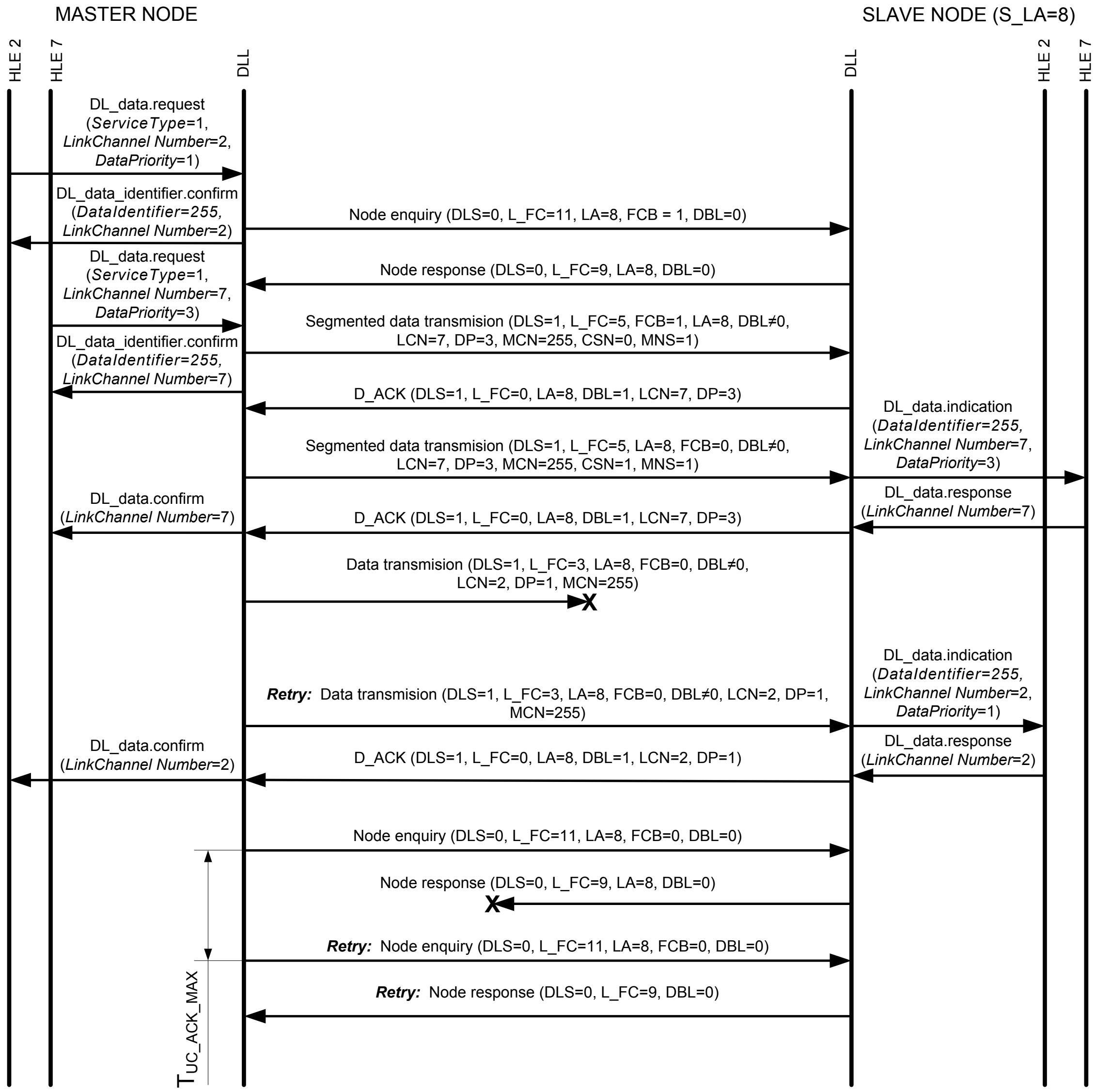

Figure 18 – Example of acknowledged unicast transmission with retry

After the expiry of time period $T_{UC\_ACK\_MAX}$ since the last unicast message to the slave node without any response/acknowledgement the master can repeat (retry) the transmission, if necessary, with another TM, FXT and FXDC values. In this case the FCB bit will not be toggled. Between retries polling and data transmission to other slave nodes continues.

If the slave node is temporarily not able to receive further messages over the control channel (DLS = 0) or one of the link-channel (DLS = 1) identified by LCN, it sets the DFC-bit in the corresponding respond/confirm message.

After expiry of the data link layer connection time-out, which is defined by the parameter S_DLLCTO, since the last unicast message to the slave node without any response/acknowledgement, the slave node will be deregistered and no longer polled by the master node.

6.5.12.2 Acknowledged multicast/broadcast transmission

An acknowledged multicast/broadcast transmission procedure can contain following steps:

- Transmission of acknowledged multicast/broadcast by a master node using ‘Data multicast’ (L_FC=4) message containing a MCN value between 0 and 199;
- Collecting of acknowledgements from the addressed slave nodes by the master node using ‘Node enquiry’(DLS=0, L_FC=11) and ‘Node response’ (DLS=0, L_FC=9) messages;
- If necessary, unicast transmission(s) of higher-layer PDUs to the slave nodes, that have not yet received the multicast/broadcast message, using ‘Data transmission’ (DLS=1, L_FC=3) message.

An acknowledged multicast/broadcast transmission procedure is completed and the multicast-buffer corresponding to an MCN value can be reused in the master node only in the following cases:

- Master node received selective acknowledgements from all addressed slave nodes by ‘Node response’ (DLS=0, L_FC=9) messages during the polling;
- Master node received selective acknowledgements from all addressed slave nodes after one or more selective (unicast based) retransmissions of the same higher layer PDU using ‘Data transmission’ (DLS=1, L_FC=3) message to the slave nodes, that had not received the PDU by ‘Data multicast’ (DLS=1, L_FC=4);
- The multicast/broadcast transmission procedure was aborted, because the master node has not received yet selective acknowledgements from all addressed slave nodes despite of one or more selective (unicast based) retransmissions of the same higher layer PDU and the time-out period T_{AMCTO} (not shown in Figure 19) expired. The T_{AMCTO} value is defined by the parameter AMCTO.

In the slave node, the corresponding multicast-buffer can be reused as soon as the acknowledgement for the reception of the multicast/broadcast message has been successfully transmitted to the master node using the flag bit MCx of the ‘Node response’ (DLS=0, L_FC=9) message (x=MCN).

Following figure shows an example of an error-free acknowledged broadcast transmission:

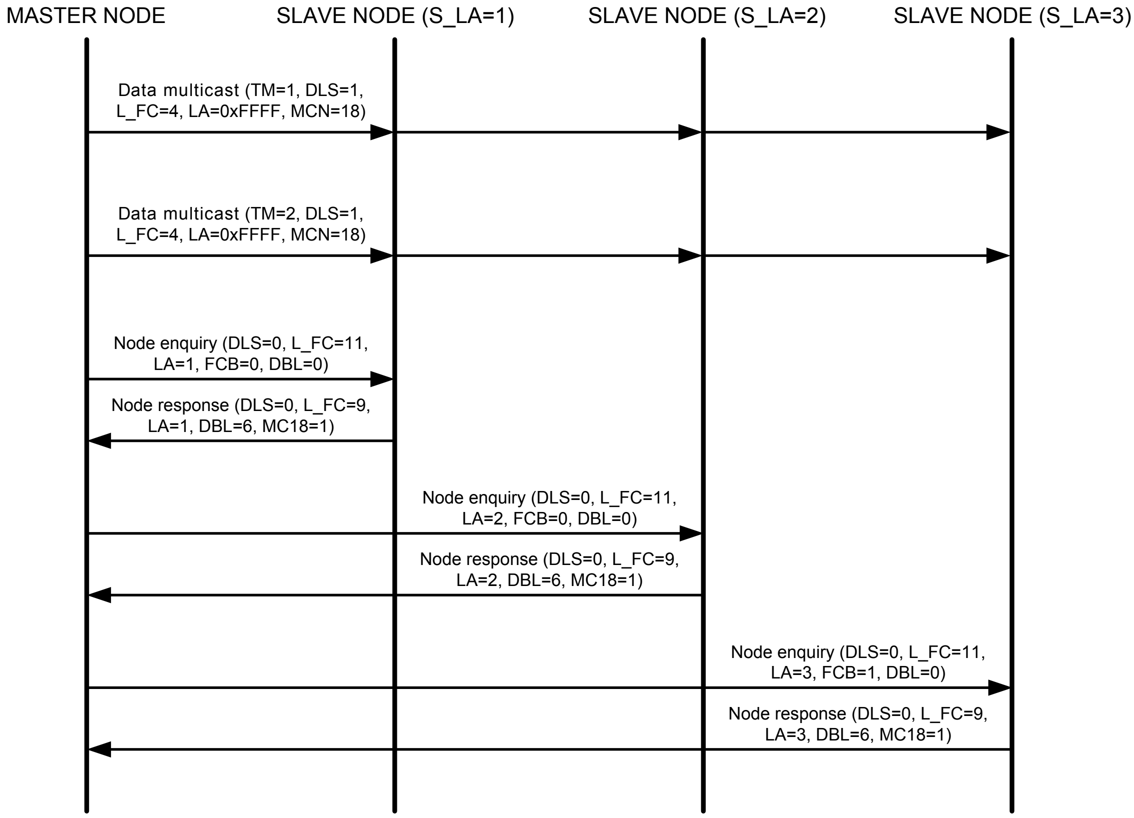

Figure 19 – Example of acknowledged multicast/broadcast transmission

As shown in the Figure 19, after a multicast transmission the master node collects the acknowledgments by polling the slave nodes using ‘Node enquiry’ (DLS=0, L_FC=11).

Multiple flag bits MC0 to MC199 may be set by the slave nodes in the ‘Node response’ (DLS=0, L_FC=9) message confirming the receptions of multicast/broadcast messages with MCN vales 0 to 199 respectively. If the transmission of a ‘Node enquiry’ or the corresponding ‘Node response’ was not error-free, the master node will repeat the transmission of the same node enquiry message with unchanged FCB value. This retry message indicates to the slave node that the previous acknowledgement(s) has not been received and that the acknowledgement(s) shall be retransmitted (see an example in Figure 20). The slave node’s response to this retry additional flag bits MC0 to MC199 may be set to confirm the receptions of further multicast/broadcast messages.

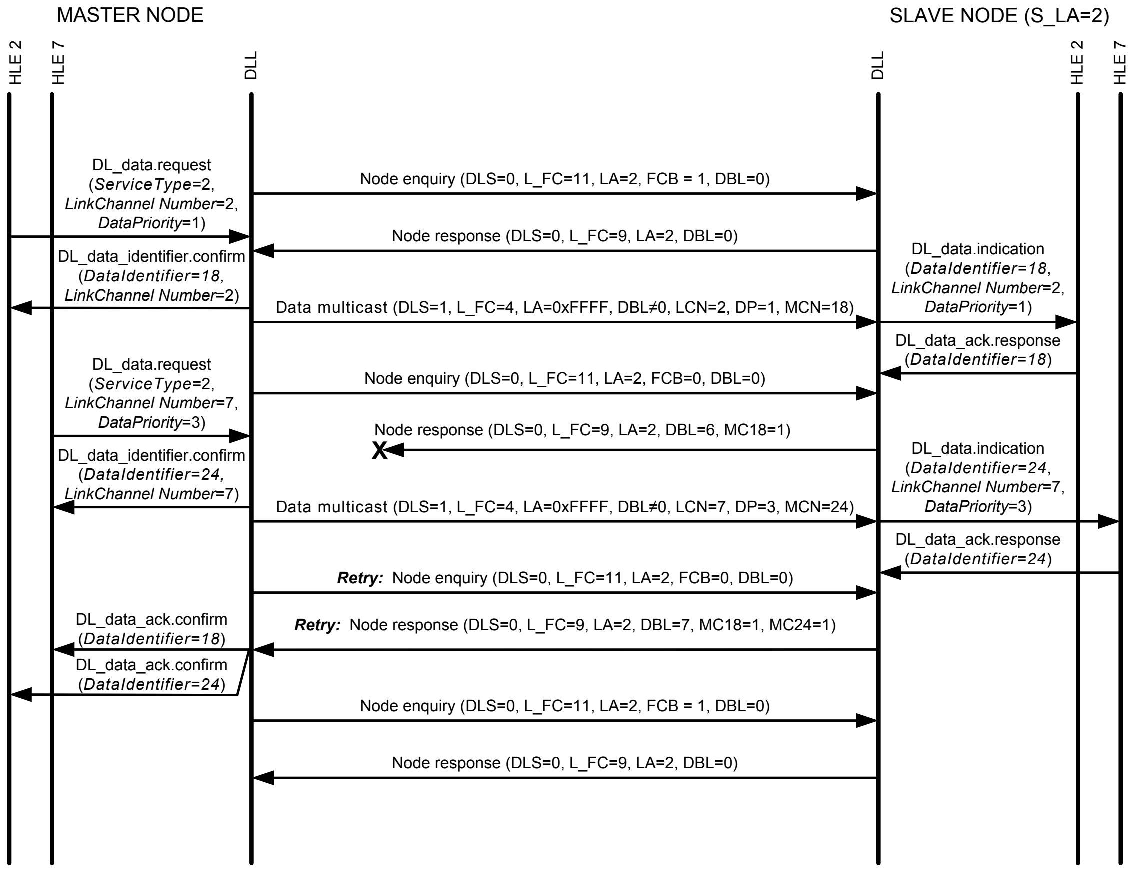

Figure 20 – Example of two multicast/broadcast transmissions with an error

The diagram in Figure 20 also shows, how both multicasts are acknowledged by the single node response.

If during the collection of acknowledgements the master node registers, that a responding slave node has not yet received the multicast/broadcast message (the corresponding flag bit MCx, x =0 to 199, was cleared in the slave node's response), then the master node will send to the slave node the same higher layer PDU by 'Data transmission' (DLS=1, L_FC=3) message (see Figure 21).

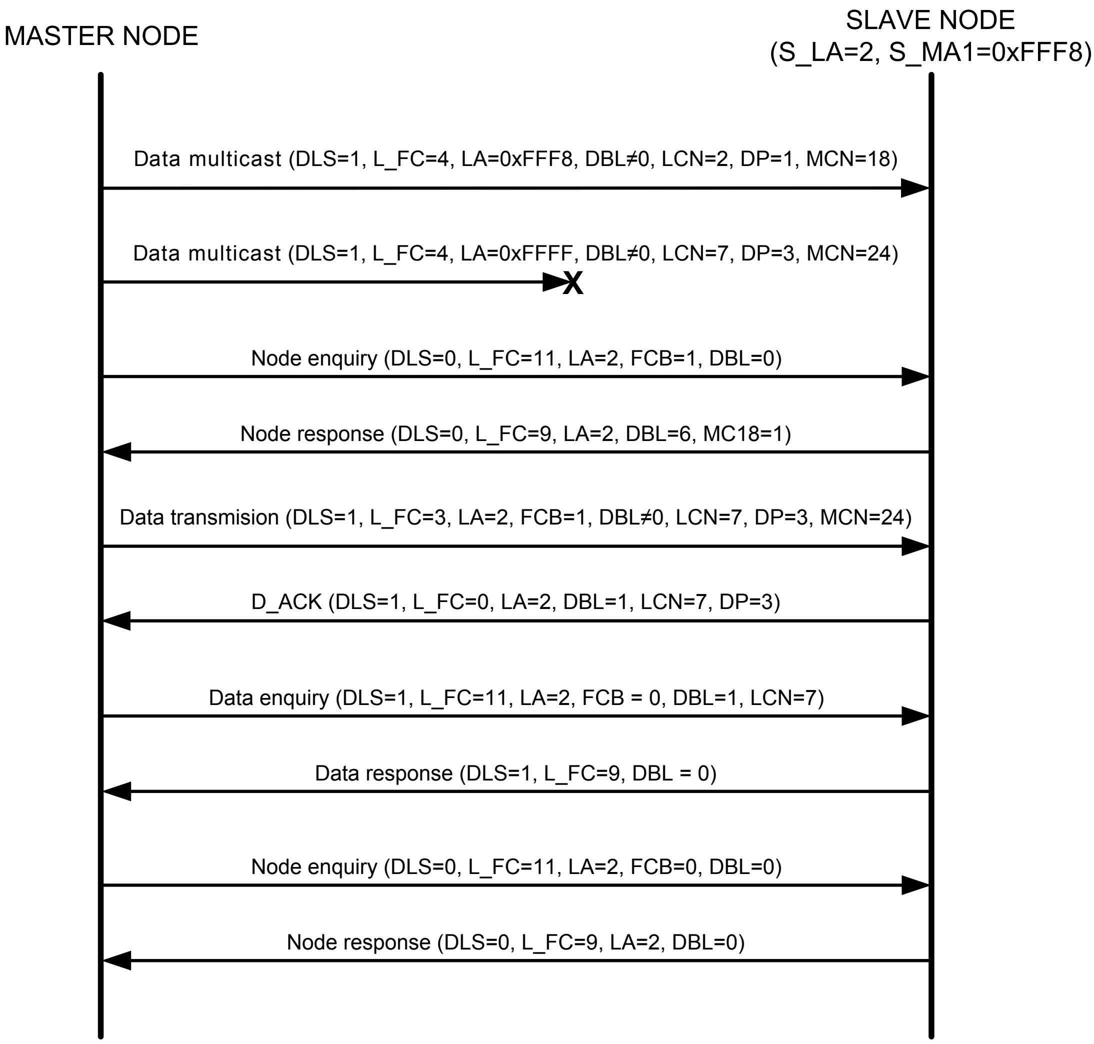

Figure 21 – Example of broadcast with message retransmission

An another reason of missing acknowledgement may be, that the slave is temporarily not able to accept further PDUs, transmitted to it over the addressed link channel. In this case the slave node signalises this by setting the DFC-bit in the ‘Node response’ (DLS=0, L_FC=9) message.

After a start-up of the master node (e.g. after a power down), the master node expects reception of acknowledgments only from those slave nodes, that are responding to ‘Node enquiry’ (DLS=0, L_FC=11) messages from the master node. Therefore multicast-frames are only transmitted to those slave nodes, which have registered with the master node after its start-up. As soon as another slave node responds to a request of the master node, it will be included in the list of nodes, that the master node is expecting acknowledgments from. If after the start-up of the master node slave nodes do not respond to its enquiry messages for the time-out period defined by the parameter M_DLLCTO, they will be deregistered in the master node.

6.5.12.3 None-acknowledged multicast/broadcast transmission

The Figure 22 shows an example of a non-acknowledged broadcast transmission procedure:

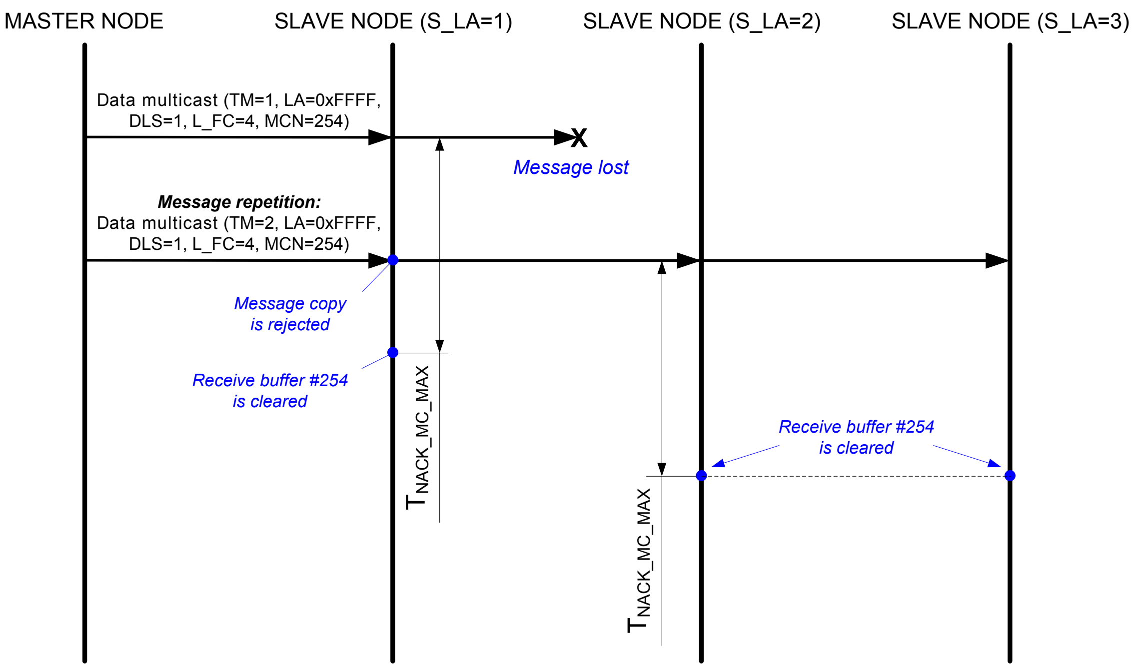

Figure 22 – Example of non-acknowledged multicast/broadcast transmission

After the finish of an unacknowledged multicast/broadcast message transmission (including frame repetitions with different TM values) the corresponding transmit buffer of the master node is free for reuse. If data from the master node consists of more than one segment, several send-confirm cycles are necessary to transmit this data to the slave node. Master node transmits the data segments in an ascending sequence of messages starting with the 1st message having CSN = 0. After a successful transmission of the last message in the sequence with CSN = MNS the transfer of the segmented data is completed.

The slave node temporarily stores the received frame to avoid duplication of the message. If the same message is received by slave node within the time period T_{MCTMAX}, it will be ignored. If a different message is received, it will be passed to the addressed higher layer entity of the slave node. The value of T_{MCTMAX} is defined by the parameter MCTMAX (cf. 6.5.11.1).

6.6 Clock synchronisation

The master node broadcasts periodically clock synchronisation information to the nodes registered with it using the 'Clock synchronisation command' (DLS=0, L_FC=4). The clock synchronisation period is defined by parameter CLCSYNCP. The broadcast message contains the local date and time (no daylight saving time) of the master node before the beginning of the frame transmission procedure.

During the whole frame forwarding procedure the time is not updated. A slave node performs message transmission delay corrections, depending on the received frame header parameter (TM, DBL, FXT, FXDC). The slave node shall use the 1st received frame during a frame forwarding procedure to set its clock. The clock in the slave node shall be buffered and continue to keep exact time during a power-down lasting longer than a time period defined by the parameter S_DLLCTO plus one hour.

6.7 Status enquiry

The message 'Status enquiry' (DLS=0, L_FC=10) can be used by a master node to check the status of the communication link to a slave node. The information contained in the 'Status response' (DLS=0, L_FC=8) can be used by the master node to optimize the transmission parameter of the communication link (e.g., TM, FXT).

6.8 PHY-link test

A master node can initiate a PHY link quality test procedure in a dedicated slave node by sending the 'Perform PHY link test' (DLS=0, L_FC=6) message as confirmed unicast (see Figure 23). Depending on the PLTM value contained in the received message, the addressed slave node transmits the 'SN PHY link test' (PRM=0, DLS=0, L_FC=7) message in the defined transmission mode. Despite PRM = 0 (direction slave node -> master node), all neighbouring slave and master nodes may receive this frame and store information about the estimated link quality (e.g receive level, SNR, bit-error-rate). As soon as the PHY link test is completed, the slave node transmits the positive acknowledgement 'ACK' (DLC=0, L_FC=0) to the master node. The master node supervises the procedure. As long as no acknowledgement is received by the master node the PHY link test procedure may be repeated by master node.

The waiting time for the acknowledgement has to take into account the duration of the PHY link test.

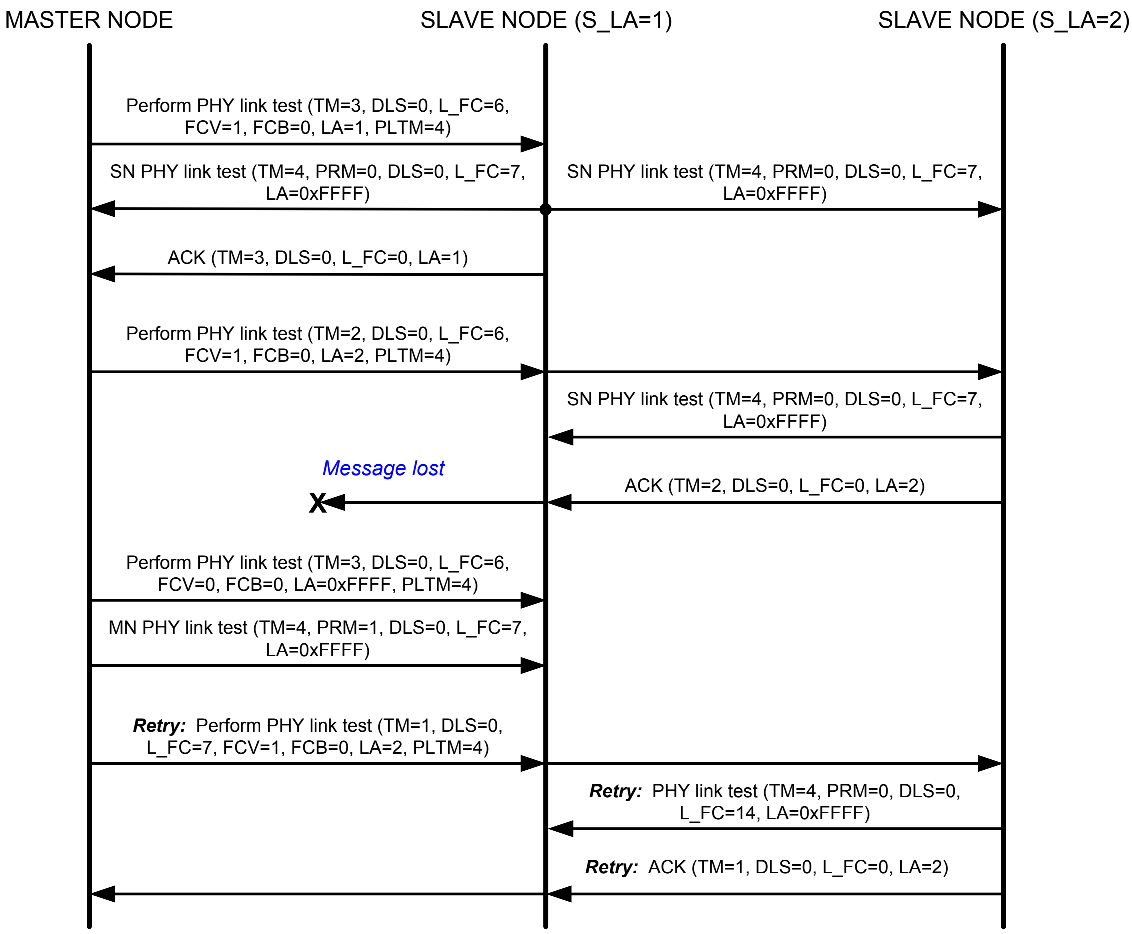

Figure 23 – Example of PHY link test

If the master node has to perform the PHY link test, it sends the message frame 'Perform PHY link test' (DLS=0, L_FC=6) within the non-confirmed multicast/broadcast first. With the message the master node signalizes to other nodes, that it is transmitting a PHY link test frame in a mode, defined by the PLTM value, immediately afterwards. Corresponding to the PLTM value the master node may transmit as second message 'MN PHY link test' (PRM=1, DLS=0, L_FC=7).

6.9 PHY quality data enquiry

Using the 'Link quality enquiry' (DLS=0, L_FC=15) message a master node can retrieve from the addressed slave node the quality information about the message reception for each hop (see Figure 24). The slave node stores the quality information for each received copy of the frame. With 'Link quality response' (DLS=0, L_FC=15) the slave transmits the collected quality information to the master node using the same TM and FXT values.

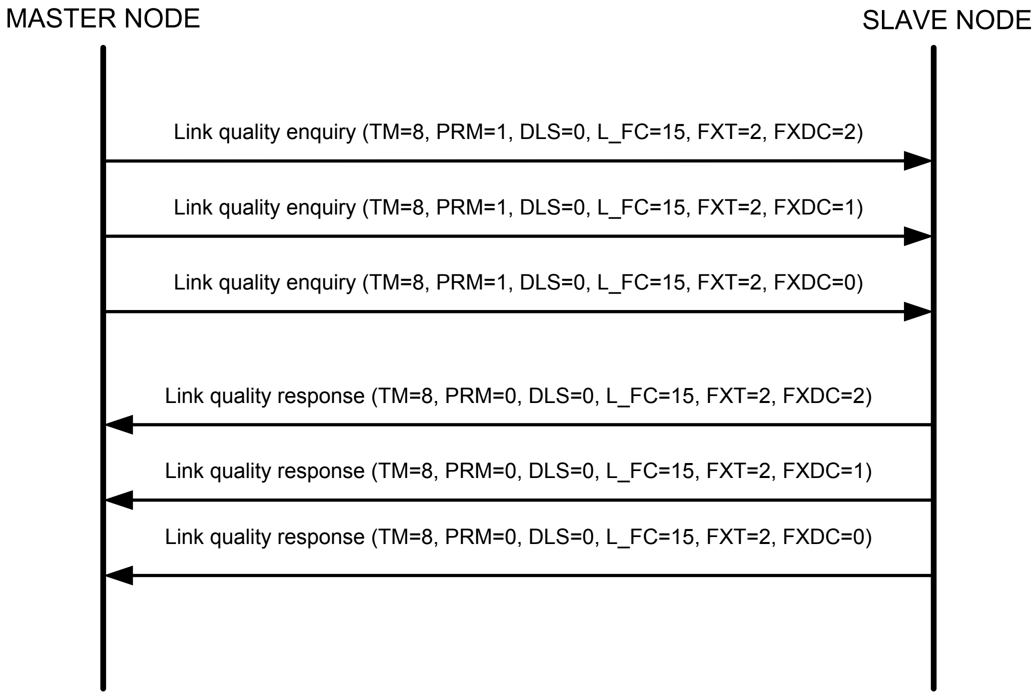

Figure 24 – Example of link quality enquiry

The master node collects the quality information for each received frame copy of the response message. The link quality test procedure can be performed with different TM values to optimize the frame transmission parameters (e.g., TM, FXT).

7 Layer-2-network capability

7.1 Overview

The data link layer uses the concept of a so-called "Network with Spreading/Shrinking Cells" (NSC) to organize communication between interconnected slave nodes (SN) and master nodes (MN) with the same value of N_NIN. Only those network nodes that are connected to a common backbone network, are used as master nodes. They use the backbone network to communicate with one or more head-end systems. The communication to and from the master nodes over the backbone network connections (BNC) is not in the scope of this specification.

A network node operating as master node and communicating with the slave nodes forms a network cell. As described in 6.3, a master node can either communicate directly with a slave node or indirectly via other slave nodes (over two or more hops). The network cell can change its form and size depending on the number of slave nodes that are registered with the master node. Cells can grow and shrink. A cell can be created, if a master node becomes active. The cell can disappear, if the master node is deactivated.

Four types of nodes are possible in a NSC:

- A primary master node (p-MN). The node is connected to the backbone network and always acts as a master node.
- A secondary master node (s-MN). The node is connected to the backbone network, normally acts as slave node, but can act as a master node, if possible.
- A relaying node (RN). The node always acts as slave node. It is used in the network (cell) only to extend the communication range of the master node to the slave nodes.
- A slave node (SN). Usually it is used for application data communication with the master node.

Within a network cell a time division multiple access is used, which is controlled by the master node polling the registered slave nodes. Frames are forwarded by the slave nodes simultaneously over several hops (cf. 6.3). Only registered slave nodes with the same S_CIN participate in regular frame forwarding in their network cell.

For the addressing of the slave nodes the MAC addressing scheme (LA, CIN and NIN) is used. The CIN and NIN, which are contained in every header, define in which cell and network the originator of the message (with the same values of M_CIN or S_CIN and N_NIN respectively) is located. During the roll-out of the NSC, all p-MN and s-MN are preconfigured with different M_CINs. A p-MN uses its M_CIN permanently. The s-MN uses its own M_CIN only if it acts as a master node. Otherwise it uses the S_CIN of the master node in the cell like a slave node. A node acting as slave node receives from the master node during a registration procedure a link address (S_LA) and a S_CIN value. All nodes of a network have the same preconfigured value of N_NIN. For neighbouring networks N_NIN values with different lower 8 bit shall be used. During the registration procedure a master node assigns to each node, acting as slave node, dynamically its S_LA, S_CIN and, if necessary, up to 31 multicast addresses S_MA1 to S_MA31. The S_CIN value corresponds to the preconfigured M_CIN value of the master node. If necessary (e. g after a firmware update), the multicast addresses may be changed by the master node during the polling. The value of M_CIN defines which communication channel is used in the network cell. In the whole network one common broadcast channel for the global (cell border crossing) broadcast with CIN=0xFFFF and LA=0xFFFF is used. Messages, which are sent over the common broadcast channel, are forwarded by all slave nodes with the same N_NIN, even if their S_CIN is different. The nodes operating as master node do not forward any messages.

An example of NSC can be found in Annex C.

7.2 Registration procedure

7.2.1 General

A registration procedure is used to assign a (new) master node to a slave node. During registration in a (new) cell, the slave node obtains its link address (S_LA), channel identification number (S_CIN) and, if necessary, one or more multicast addresses (S_MAx, x=1 to 31) from the master node.

7.2.2 Registration of a new slave node

Master node performs the registration procedure periodically in the background. If a node has not yet been registered or was deregistered with the master node, it uses S_LA=0xFFFF and S_CIN=0xFFFF.

The registration procedure of a slave node by a master node consists of following steps:

- Master node broadcasts a message ‘Generate temporary link address’ (DLS=0, L_FC=0).
- Slave nodes that have not yet been registered with the master node, generate a random value of temporary link address (TLA) according to the RTLA value from the received message.
- The master node sends sequentially ‘Request to node with temporary link address’ (DLS=0, L_FC=8) with different TLA values in the range from 0 to RTLA, starting with the TLA=0.
- A slave node, which has a matching TLA, responds with the message ‘Response from node with temporary link address’ (DLS=0, L_FC=10) transmitting to the master node especially its device identifier (DID) and information about the used data link channels.
- Using the message ‘Assign new link address’ (DLS=0, L_FC=9) the master node assigns to the slave node new S_LA, S_CIN and, if necessary, up to 31 multicast addresses S_MA1 to S_MA31. If several nodes have generated the same random TLA values, a frame collision may occur during the response phase and the nodes will not be registered in this registration attempt.
- After the master node has polled all possible temporary link addresses, a new registration procedure will be started.

An example of the registration procedure is shown in Figure 25.

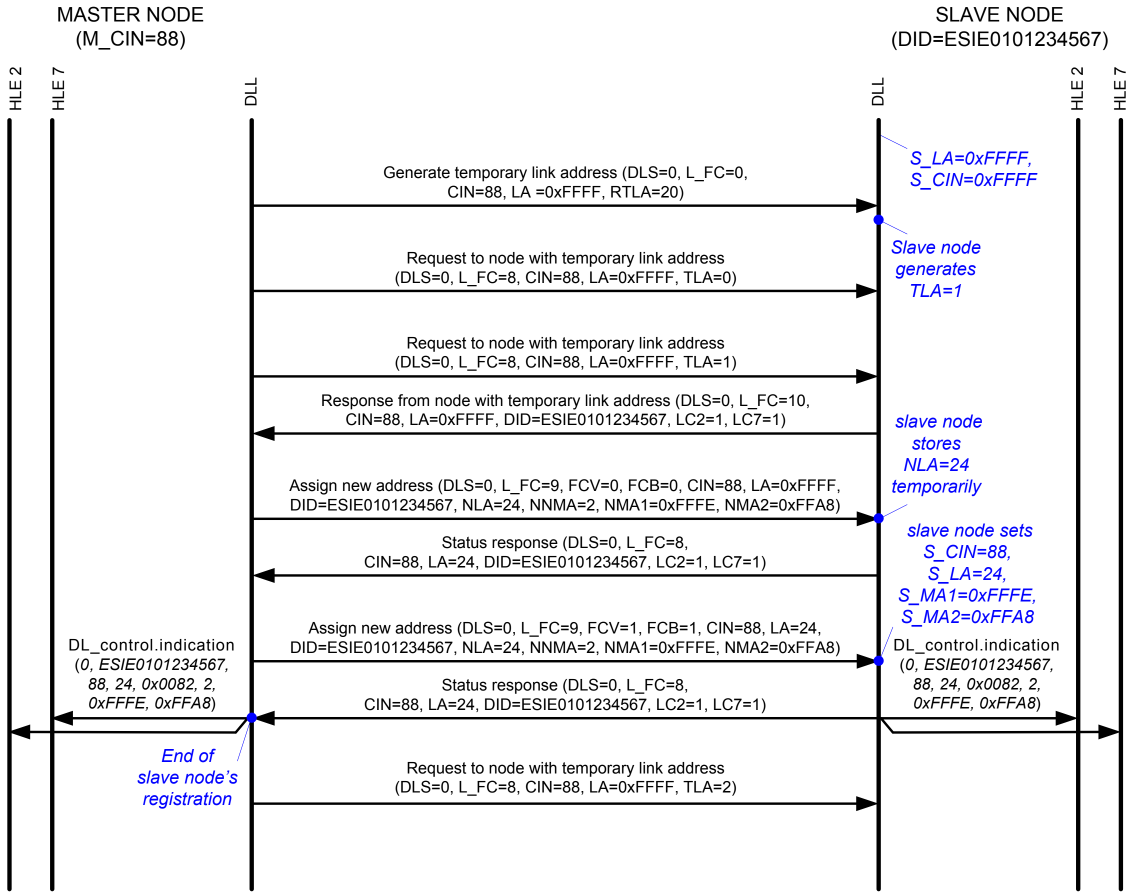

Figure 25 – Example of slave node registration procedure

The message ‘Assign new link address’ (DLS=0, L_FC=9) is sent by the master node at least two times:

- 1st message as broadcast with LA=0xFFFF, FCV=0, FCB=0. After receiving of the 1st message, the slave node stores the received CIN, new link address (NLA) and, if available, new multicast addresses (NMAx) temporarily until the reception of the 2nd message from the master, which is described below, or restart of the registration procedure by a master node. With regard to FCB bit this message has the same function as the message ‘Clear transmit buffer’ (DLS=0, L_FC=9).

- 2nd message is send as unicast with FCV=1, FCB=1 and LA=NLA. The master node sends this message after successful receiving of the ‘Status response’ (DLS=0, L_FC=9) from the slave node as response to the 1st master node’s message. After receiving of the 2nd message, the slave node accepts the temporarily stored CIN, NLA and NMAx as S_CIN, S_LA and S_MAx respectively.

Until the 1st or 2nd message has not been successfully received by the slave node or the slave node’s responses by the master node, the master node may run retries several times by resending the 1st or 2nd message respectively. During the retries the FCB bit in the message is not alternated by the master node.

As shown in Figure 26, the message ‘Assign new link address’ (DLS=0, L_FC=9) may be used by master node to assign new or additional multicast addresses S_MAx, x = 1 to 31) or delete all or some of them in a slave node, which is registered with the master node. The value of NLA in the message shall be the same as the S_LA value of the slave node. Otherwise it ignores the message.

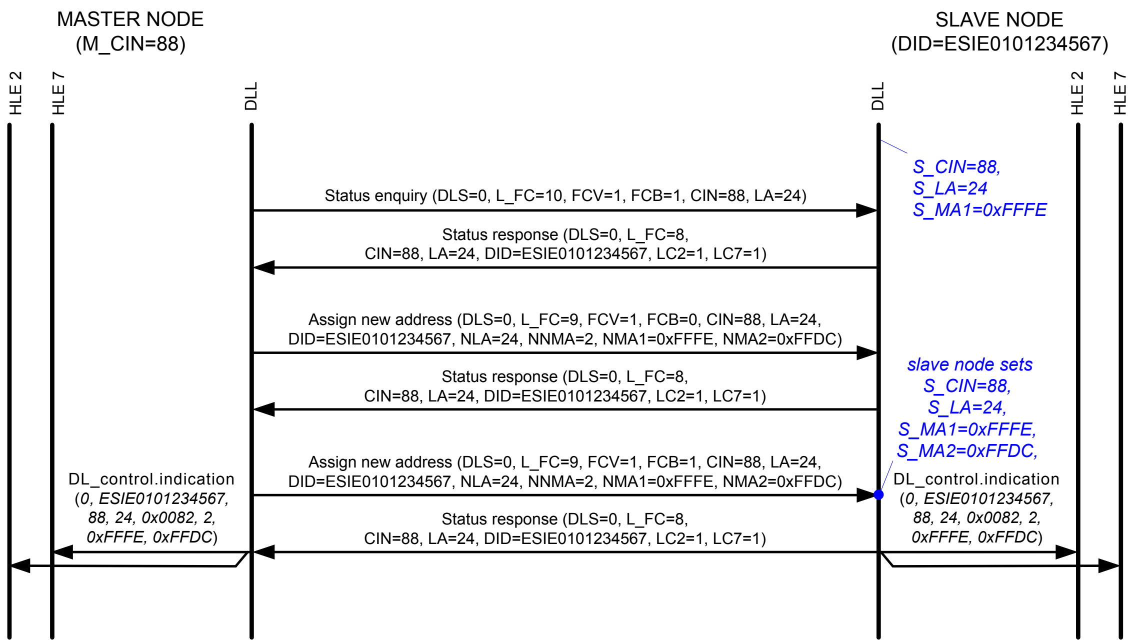

Figure 26 – Example of new multicast address assignment

An successful registration or deregistration of the slave node, its temporary inaccessibility by the master node as well as assignment of new multicast addresses are indicated by a DL_control.indication primitive to all higher layer entities of the slave node and the master node.

7.2.3 Data link connection time-out

Supervision of communication

If a master node does not receive any responses from a registered slave node for a time period longer than a value defined by the parameter M_DLLCTO, it deregisters the slave node (erases its DID, LA and, if necessary the corresponding MAx, x=0 to 31 from the list of the registered and polled slave nodes).

A registered slave node will clear its address parameter (set S_LA, S_CIN and S_MAx, x=0 to 31 to 0xFFFF), if no unicast message from the master node is received for a time period longer than the time-out period defined by the parameter S_DLLCTO:

7.2.4 Re-establishing of data link connection after power-down

If a power-down is detected in a registered slave node, it saves its S_CIN, S_LA and S_MAx together with the current time stamp in a non-volatile memory.

After the end of a power-down period the slave node determines its duration using the internal clock verified by the master node's message 'Clock synchronization command' (DLS=0, L_FC=4), which has kept an exact time during the whole power-down period. The power-down period T_{SNPD} is determined as the time period starting with the time stamp of saved addresses till the end of the last power-down period. Depending on the T_{SNPD} value following actions are possible:

- If T_{SNPD} < S_PDDTH, the slave node restore its S_CIN, S_LA and S_MAx with the values saved in a non-volatile memory and continue to operate as it had operated before power-down;

- If T_{SNPD} ≥ S_PDDTH, the slave node restores first only its S_CIN, from the non-volatile memory and sets its S_LA=0xFFFE and S_MAx=0xFFFF, x=0 to 31 (wait-state link-address value). The wait-state link address cannot be assigned by a master node. In this state the slave node is waiting for the 'Reuse link address' (DLS=0, L_FC=5) message from the master node, it is registered with. If the time point in a received 'Reuse link address' message is earlier than the time stamp stored in the slave node, it restores remaining the values of S_LA and S_MAx, x=0 to 31 from its non-volatile memory and continues to use them as before the power-down. If the time point in a received 'Reuse link address' message is not earlier than the time stamp stored in the slave node, or if the slave node is not receiving the message 'Reuse link address' within the time period defined by the parameter S_DLLCTO since its power-up or, it discards its current address information and sets S_LA, S_CIN and S_MAx, x=0 to 31 to 0xFFFF. Supposing that its last registration with the master node got lost, the slave node will try to register again with a master node as described in 7.2.2.

On the other side, if one or more slave nodes have not been responded to the master node at least for a time period defined by the parameter DLLCRSTBTH, the master node start to broadcasts periodically for the duration defined by the parameter M_DLLCTO the message 'Reuse link address' (DLS=0, L_FC=5) in order to re-establish the missing DLL connections to the slave nodes. Those slave nodes, that during this time period have not responded to master node's enquiries or became registered with it using the registration procedure, are removed from the master node's registration list.

If a power-down is detected in a master node, it saves all the information, which is necessary to recover communication with the registered slave nodes, together with the current time stamp in a non-volatile memory.

After a restart from a power-down the master node tries to re-establish the communication to the slave nodes that it still has had registered. If at least one of the slave nodes does not respond, the master node start to broadcasts periodically for the duration defined by the parameter M_DLLCTO the message ‘Reuse link address’ (DLS=0, L_FC=5). This message may be only sent, if the time period since the last power-down of the master node is not longer than the value defined by the parameter M_PDDTH. Those slave nodes, that during this time period have not responded to the master node’s enquiries or became registered with it using the registration procedure, are removed from the registration list of the master node.

7.3 Coordination of master nodes

Each network node, which is operating as master node, broadcasts periodically and asynchronously to other master nodes the message ‘Master node beacon’ (DLS=0, L_FC=2) with FXS = 0xF, CIN = 0xFFFF and LA = 0xFFFF. The beacon repetition period is defined by the parameter BRP. The message may cross the boundary of the cell around the master. This broadcast will be forwarded simultaneously by all slave nodes of the network. Other active master nodes may only receive this broadcast, but will not forward it.

By receiving such beacon messages a s-MN can determine the distance to the originating master node in hops and estimate the size of its cell. Depending on the distance, the s-MN may start or stop to operate as master node.

E. g., the s-MN operating as a slave node may become a master node, in one of the following cases:

- S-MN has not received any beacon message from a master node with the same N_NIN during a beacon time-out period defined by the parameter BTO;
- S-MN has received beacon messages from a least one master node with the same N_NIN during the beacon time-out period (BTO) and the distance to the neighbouring master node during a beacon monitoring period defined by the parameter BMWT was bigger than a threshold value (e.g. 5 hops) defined by the parameter MNACTTH.

When the s-MN becomes active, a new cell is formed around this node.

If, for example, the distance between an s-MN operating as master node and one p-MN is not bigger than a value (e.g. 4 hops) defined by the parameter MNDEACTTH, the s-MN begins to become inactive. In this case the s-MN broadcasts the message ‘Master node deactivated’ (DLS = 0, L_FC = 3) to the slave nodes, that are registered with it, and discontinues sending of ‘Master node beacon’ messages. After all its slave nodes are deregistered but at latest after the time period defined by the parameter S_DLLCTO, the s-MN stops to operate as master node becoming a slave node and retries to register with another master node.

This s-MN behaviour provides a high accessibility of slave nodes. Presence of several redundant s-MNs improves availability of a network.

Examples of an s-MN becoming active are given in Annex C.

7.4 Cell change by slave node

If a slave node does no longer receive messages from its master node, it will search for alternative master nodes within the same network by evaluating messages from the neighbouring cells (without relaying those). Even if the slave node receives messages from its master node, it may from time to time evaluate messages from other cells in order to find out if a better connection to an alternative master node exists. As a criterion for that the frame reception quality and/or the distance in hops could be used. The slave node's cell-change options may be limited, if the slave node has received a list with one ore more M_CIN of allowed master nodes. If this list contains M_CIN of only one master the slave will always try to register with this master node. If the list is empty all other master nodes are allowed and the slave node may try to change its current cell (to register with another master node). Every master node performs in the background search for new slave nodes and their registration (cf. section 7.2.2). This is done by broadcasting the frames 'Generate temporary link address' (DLS=0, L_FC=0) and 'Request to node with temporary link address' (DLS=0, L_FC=8). Especially the messages 'Request to node with temporary link address' are used by the slave node to analyse a suitability of a connection to an alternative master node.

If a new alternative master node with better communication quality has been found within the same network, the slave node will participate in the slave registration procedure of the new master node (see an example in Figure 27 and cf. section 7.2.2). The slave node keeps its registration on the current master none as long as it takes to become registered with the new master node. After a successful assignment of the new NLA and, if available, NMAx, x=0 to 31 by the new master node, the slave node sets S_LA, S_CIN and S_MAx, x=0 to 31 to the new values and clears the information of its registration with the previous master node.

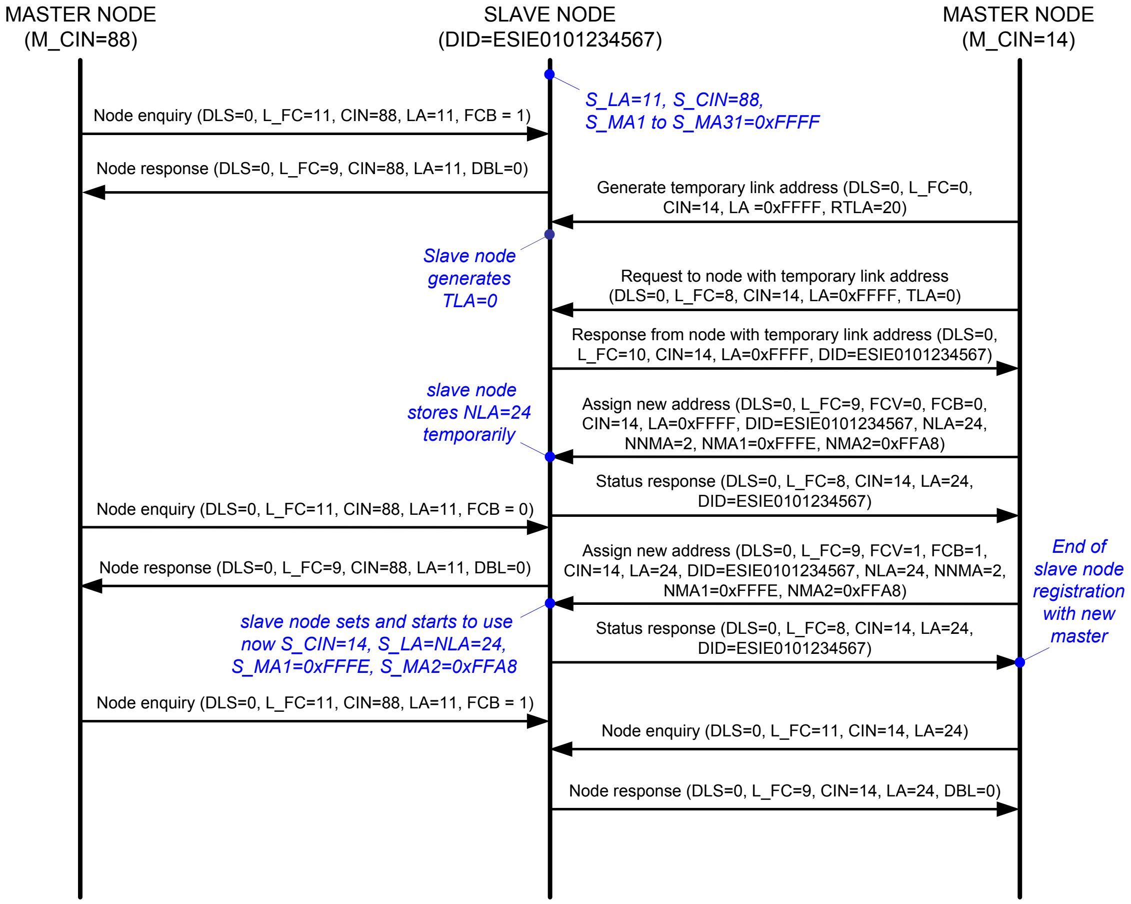

Figure 27 – Example of cell change by slave node

The previous master node is going to deregister the slave node (erase its DID, LA and, if necessary the corresponding MAx, x=0 to 31 from the list of the registered and polled slave nodes) at latest after the time period defined by the parameter S_DLLCTO.

The capability of the slave nodes to search for better communication options results in a permanent adaptation and optimisation of the communication network.

Annex A
(normative)

A.1 Window functions

The following Table A.1 to Table A.6 define the sample values w(i) of the window functions W0, W1, W2, W3, W4 and W5 with the sample interval $\Delta t_{Tx} = 1/f_{Tx}$ = 347200 samples/s (cf. 5.3.6). The values w0(i), w1(i), w4(i) and w5(i) are given as two‘s complement numbers in 1.15 fractional format and hexadecimal notation.

Table A.1 – Samples of the window functions W0

i	w0(i)	i	w0(i)	i	w0(i)	i	w0(i)	i	w0(i)
1	0x01E7	32	0x2C4E	63	0x73EA	94	0x708A	125	0x27D3
2	0x0260	33	0x2E9C	64	0x7578	95	0x6EB9	126	0x25A8
3	0x02E5	34	0x30F6	65	0x76EC	96	0x6CD4	127	0x238B
4	0x0378	35	0x3358	66	0x7848	97	0x6ADB	128	0x217B
5	0x0417	36	0x35C4	67	0x7989	98	0x68D0	129	0x1F7B
6	0x04C5	37	0x3837	68	0x7AB0	99	0x66B4	130	0x1D89
7	0x0581	38	0x3AB1	69	0x7BBB	100	0x6487	131	0x1BA6
8	0x064C	39	0x3D30	70	0x7CAA	101	0x624C	132	0x19D4
9	0x0727	40	0x3FB5	71	0x7D7D	102	0x6003	133	0x1811
10	0x0810	41	0x423D	72	0x7E32	103	0x5DAE	134	0x165F
11	0x090A	42	0x44C7	73	0x7ECA	104	0x5B4E	135	0x14BE
12	0x0A14	43	0x4753	74	0x7F44	105	0x58E4	136	0x132D
13	0x0B2E	44	0x49DF	75	0x7F9F	106	0x5671	137	0x11AD
14	0x0C59	45	0x4C69	76	0x7FDD	107	0x53F7	138	0x103F
15	0x0D95	46	0x4EF2	77	0x7FFB	108	0x5177	139	0x0EE1
16	0x0EE1	47	0x5177	78	0x7FFB	109	0x4EF2	140	0x0D95
17	0x103F	48	0x53F7	79	0x7FDD	110	0x4C69	141	0x0C59
18	0x11AD	49	0x5671	80	0x7F9F	111	0x49DF	142	0x0B2E
19	0x132D	50	0x58E4	81	0x7F44	112	0x4753	143	0x0A14
20	0x14BE	51	0x5B4E	82	0x7ECA	113	0x44C7	144	0x090A
21	0x165F	52	0x5DAE	83	0x7E32	114	0x423D	145	0x0810
22	0x1811	53	0x6003	84	0x7D7D	115	0x3FB5	146	0x0727
23	0x19D4	54	0x624C	85	0x7CAA	116	0x3D30	147	0x064C
24	0x1BA6	55	0x6487	86	0x7BBB	117	0x3AB1	148	0x0581
25	0x1D89	56	0x66B4	87	0x7AB0	118	0x3837	149	0x04C5
26	0x1F7B	57	0x68D0	88	0x7989	119	0x35C4	150	0x0417
27	0x217B	58	0x6ADB	89	0x7848	120	0x3358	151	0x0378
28	0x238B	59	0x6CD4	90	0x76EC	121	0x30F6	152	0x02E5
29	0x25A8	60	0x6EB9	91	0x7578	122	0x2E9C	153	0x0260
30	0x27D3	61	0x708A	92	0x73EA	123	0x2C4E	154	0x01E7
31	0x2A0A	62	0x7246	93	0x7246	124	0x2A0A		

Table A.2 – Samples of the window functions W1

i	w1(i)	i	w1(i)	i	w1(i)	i	w1(i)	i	w1(i)
1	0x01A5	31	0x3735	61	0x7EE8	91	0x4CBC	121	0x06E7
2	0x022B	32	0x3A38	62	0x7F70	92	0x49A2	122	0x05E8
3	0x02C4	33	0x3D43	63	0x7FCC	93	0x4687	123	0x0500
4	0x036F	34	0x4055	64	0x7FF9	94	0x436C	124	0x042D
5	0x042D	35	0x436C	65	0x7FF9	95	0x4055	125	0x036F
6	0x0500	36	0x4687	66	0x7FCC	96	0x3D43	126	0x02C4
7	0x05E8	37	0x49A2	67	0x7F70	97	0x3A38	127	0x022B
8	0x06E7	38	0x4CBC	68	0x7EE8	98	0x3735	128	0x01A5
9	0x07FC	39	0x4FD2	69	0x7E32	99	0x343E		
10	0x0928	40	0x52E3	70	0x7D51	100	0x3153		
11	0x0A6D	41	0x55EB	71	0x7C44	101	0x2E77		
12	0x0BC9	42	0x58E9	72	0x7B0C	102	0x2BAA		
13	0x0D3F	43	0x5BDA	73	0x79AB	103	0x28EF		
14	0x0ECD	44	0x5EBD	74	0x7823	104	0x2646		
15	0x1075	45	0x618E	75	0x7673	105	0x23B0		
16	0x1236	46	0x644B	76	0x749E	106	0x2130		
17	0x1410	47	0x66F2	77	0x72A5	107	0x1EC5		
18	0x1604	48	0x6982	78	0x708B	108	0x1C71		
19	0x1810	49	0x6BF7	79	0x6E50	109	0x1A34		
20	0x1A34	50	0x6E50	80	0x6BF7	110	0x1810		
21	0x1C71	51	0x708B	81	0x6982	111	0x1604		
22	0x1EC5	52	0x72A5	82	0x66F2	112	0x1410		
23	0x2130	53	0x749E	83	0x644B	113	0x1236		
24	0x23B0	54	0x7673	84	0x618E	114	0x1075		
25	0x2646	55	0x7823	85	0x5EBD	115	0x0ECD		
26	0x28EF	56	0x79AB	86	0x5BDA	116	0x0D3F		
27	0x2BAA	57	0x7B0C	87	0x58E9	117	0x0BC9		
28	0x2E77	58	0x7C44	88	0x55EB	118	0x0A6D		
29	0x3153	59	0x7D51	89	0x52E3	119	0x0928		
30	0x343E	60	0x7E32	90	0x4FD2	120	0x07FC		

Table A.3 – Samples of the window functions W2

i	w2(i)	i	w2(i)	i	w2(i)	i	w2(i)
1	0,00015059	33	0,51227061	65	0,99984941	97	0,48772939
2	0,00135477	34	0,53678228	66	0,99864523	98	0,46321772
3	0,00376023	35	0,56120534	67	0,99623977	99	0,43879466
4	0,00736118	36	0,58548094	68	0,99263882	100	0,41451906
5	0,01214893	37	0,60955062	69	0,98785107	101	0,39044938
6	0,01811197	38	0,63335638	70	0,98188803	102	0,36664362
7	0,02523591	39	0,65684087	71	0,97476409	103	0,34315913
8	0,0335036	40	0,67994752	72	0,9664964	104	0,32005248
9	0,04289512	41	0,70262066	73	0,95710488	105	0,29737934
10	0,05338785	42	0,72480566	74	0,94661215	106	0,27519434
11	0,0649565	43	0,7464491	75	0,9350435	107	0,2535509
12	0,07757322	44	0,76749881	76	0,92242678	108	0,23250119
13	0,09120759	45	0,7879041	77	0,90879241	109	0,2120959
14	0,10582679	46	0,8076158	78	0,89417321	110	0,1923842
15	0,12139558	47	0,82658642	79	0,87860442	111	0,17341358
16	0,13787646	48	0,84477027	80	0,86212354	112	0,15522973
17	0,15522973	49	0,86212354	81	0,84477027	113	0,13787646
18	0,17341358	50	0,87860442	82	0,82658642	114	0,12139558
19	0,1923842	51	0,89417321	83	0,8076158	115	0,10582679
20	0,2120959	52	0,90879241	84	0,7879041	116	0,09120759
21	0,23250119	53	0,92242678	85	0,76749881	117	0,07757322
22	0,2535509	54	0,9350435	86	0,7464491	118	0,0649565
23	0,27519434	55	0,94661215	87	0,72480566	119	0,05338785
24	0,29737934	56	0,95710488	88	0,70262066	120	0,04289512
25	0,32005248	57	0,9664964	89	0,67994752	121	0,0335036
26	0,34315913	58	0,97476409	90	0,65684087	122	0,02523591
27	0,36664362	59	0,98188803	91	0,63335638	123	0,01811197
28	0,39044938	60	0,98785107	92	0,60955062	124	0,01214893
29	0,41451906	61	0,99263882	93	0,58548094	125	0,00736118
30	0,43879466	62	0,99623977	94	0,56120534	126	0,00376023
31	0,46321772	63	0,99864523	95	0,53678228	127	0,00135477
32	0,48772939	64	0,99984941	96	0,51227061	128	0,00015059

Table A.4 – Samples of the window functions W3

i	w3(i)
1	0,0092581
2	0,0366894
3	0,0812781
4	0,141373
5	0,2147487
6	0,2986879
7	0,3900821
8	0,4855468
9	0,5815468
10	0,6745269
11	0,7610438
12	0,8378938
13	0,9022308
14	0,9516722
15	0,9843873
16	0,9991644
17..163	1,0000000
164	0,99074195
165	0,96331063
166	0,91872189
167	0,85862695
168	0,78525126
169	0,70131209
170	0,60991788
171	0,51445317
172	0,41845322
173	0,32547314
174	0,23895617
175	0,16210623
176	0,09776925
177	0,04832776
178	0,0156127
179	0,00083558

Table A.5 – Samples of the window functions W4

i	w4(i)	i	w4(i)	i	w4(i)	i	w4(i)	i	w4(i)
1	0x00CD	31	0x2D05	61	0x7C1F	91	0x54B2	121	0x0A14
2	0x011E	32	0x2FDC	62	0x7D36	92	0x519C	122	0x08E4
3	0x017C	33	0x32C2	63	0x7E20	93	0x4E7F	123	0x07CA
4	0x01E9	34	0x35B7	64	0x7EDD	94	0x4B5E	124	0x06C7
5	0x0265	35	0x38B9	65	0x7F6B	95	0x483A	125	0x05DB
6	0x02F1	36	0x3BC5	66	0x7FCA	96	0x4517	126	0x0503
7	0x038F	37	0x3EDA	67	0x7FF9	97	0x41F6	127	0x043F
8	0x043F	38	0x41F6	68	0x7FF9	98	0x3EDA	128	0x038F
9	0x0503	39	0x4517	69	0x7FCA	99	0x3BC5	129	0x02F1
10	0x05DB	40	0x483A	70	0x7F6B	100	0x38B9	130	0x0265
11	0x06C7	41	0x4B5E	71	0x7EDD	101	0x35B7	131	0x01E9
12	0x07CA	42	0x4E7F	72	0x7E20	102	0x32C2	132	0x017C
13	0x08E4	43	0x519C	73	0x7D36	103	0x2FDC	133	0x011E
14	0x0A14	44	0x54B2	74	0x7C1F	104	0x2D05	134	0x00CD
15	0x0B5D	45	0x57BF	75	0x7ADD	105	0x2A40		
16	0x0CBE	46	0x5AC0	76	0x796F	106	0x278E		
17	0x0E38	47	0x5DB3	77	0x77D8	107	0x24F0		
18	0x0FCC	48	0x6095	78	0x761A	108	0x2267		
19	0x1178	49	0x6364	79	0x7435	109	0x1FF4		
20	0x133E	50	0x661E	80	0x722B	110	0x1D99		
21	0x151E	51	0x68C0	81	0x6FFF	111	0x1B55		
22	0x1717	52	0x6B47	82	0x6DB2	112	0x192A		
23	0x192A	53	0x6DB2	83	0x6B47	113	0x1717		
24	0x1B55	54	0x6FFF	84	0x68C0	114	0x151E		
25	0x1D99	55	0x722B	85	0x661E	115	0x133E		
26	0x1FF4	56	0x7435	86	0x6364	116	0x1178		
27	0x2267	57	0x761A	87	0x6095	117	0x0FCC		
28	0x24F0	58	0x77D8	88	0x5DB3	118	0x0E38		
29	0x278E	59	0x796F	89	0x5AC0	119	0x0CBE		
30	0x2A40	60	0x7ADD	90	0x57BF	120	0x0B5D		

Table A.6 – Samples of the window functions W5

i	w5(i)	i	w5(i)	i	w5(i)	i	w5(i)
1	0x0203	31	0x5525	61	0x6DB6	91	0x0A6C
2	0x02CD	32	0x5900	62	0x6AA3	92	0x08C8
3	0x03B8	33	0x5CC5	63	0x6763	93	0x074C
4	0x04C5	34	0x6070	64	0x63FB	94	0x05F6
5	0x05F6	35	0x63FB	65	0x6070	95	0x04C5
6	0x074C	36	0x6763	66	0x5CC5	96	0x03B8
7	0x08C8	37	0x6AA3	67	0x5900	97	0x02CD
8	0x0A6C	38	0x6DB6	68	0x5525	98	0x0203
9	0x0C39	39	0x7097	69	0x513B		
10	0x0E30	40	0x7344	70	0x4D44		
11	0x1050	41	0x75B7	71	0x4947		
12	0x129B	42	0x77EE	72	0x4548		
13	0x150F	43	0x79E6	73	0x414B		
14	0x17AC	44	0x7B9B	74	0x3D55		
15	0x1A72	45	0x7D0B	75	0x396B		
16	0x1D60	46	0x7E34	76	0x3590		
17	0x2073	47	0x7F14	77	0x31C8		
18	0x23AB	48	0x7FAA	78	0x2E17		
19	0x2706	49	0x7FF6	79	0x2A7F		
20	0x2A7F	50	0x7FF6	80	0x2706		
21	0x2E17	51	0x7FAA	81	0x23AB		
22	0x31C8	52	0x7F14	82	0x2073		
23	0x3590	53	0x7E34	83	0x1D60		
24	0x396B	54	0x7D0B	84	0x1A72		
25	0x3D55	55	0x7B9B	85	0x17AC		
26	0x414B	56	0x79E6	86	0x150F		
27	0x4548	57	0x77EE	87	0x129B		
28	0x4947	58	0x75B7	88	0x1050		
29	0x4D44	59	0x7344	89	0x0E30		
30	0x513B	60	0x7097	90	0x0C39		

A.2 Bit interleaving schemes

The bit interleaving schemes are contained in Table A.7 to Table A.26. The rows and columns of the bit interleaving schemes contain additional headings. In the headings of the columns b0 to b2 are bit inputs of the DPSK mapper (cf. Table 5, Table 6 and Table 7) and f_j, j=5,7,8,9,16,20 are the carrier frequencies from the corresponding FMS (see Table 2). The headings of the rows contain the number of symbol or FH cycle.

Table A.7 – Bit interleaving scheme for TM0 and PHY header

FH cycle	f_1	f_4	f_7	f_2	f_5	f_8	f_3	f_6
	b_0	b_0	b_0	b_0	b_0	b_0	b_0	b_0
0	0	48	1	49	2	50	3	51
1	4	52	5	53	6	54	7	55
2	8	56	9	57	10	58	11	59
3	12	60	13	61	14	62	15	63
4	16	64	17	65	18	66	19	67
5	20	68	21	69	22	70	23	71
6	24	72	25	73	26	74	27	75
7	38	76	39	77	40	78	41	79
8	32	80	33	81	34	82	35	83
9	36	84	37	85	38	86	39	87
10	40	88	41	89	42	90	43	91
11	44	92	45	93	46	94	47	95

Table A.8 – Bit interleaving scheme for TM1

FH cycle	f_1	f_3	f_5	f_2	f_4
	b_0	b_0	b_0	b_0	b_0
0	0	60	1	61	2
1	30	90	31	91	32
2	62	3	63	4	64
3	92	33	93	34	94
4	5	65	6	66	7
5	35	95	36	96	37
6	67	8	68	9	69
7	97	38	98	39	99
8	10	70	11	71	12
9	40	100	41	101	42
10	72	13	73	14	74
11	102	43	103	44	104
12	15	75	16	76	17
13	45	105	46	106	47
14	77	18	78	19	79
15	107	48	108	49	109
16	20	80	21	81	22
17	50	110	51	111	52
18	82	23	83	24	84
19	112	53	113	54	114
20	25	85	26	86	27
21	55	115	56	116	57
22	87	28	88	29	89
23	117	58	118	59	119

Table A.9 – Bit interleaving scheme for TM2

FH cycle	f_1		f_4		f_7		f_2		f_5		f_8		f_3		f_6	
	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1
0	0	48	1	49	2	50	3	51	4	52	5	53	6	54	7	55
1	96	144	97	145	98	146	99	147	100	148	101	149	102	150	103	151
2	8	56	9	57	10	58	11	59	12	60	13	61	14	62	15	63
3	104	152	105	153	106	154	107	155	108	156	109	157	110	158	111	159
4	16	64	17	65	18	66	19	67	20	68	21	69	22	70	23	71
5	112	160	113	161	114	162	115	163	116	164	117	165	118	166	119	167
6	24	72	25	73	26	74	27	75	28	76	29	77	30	78	31	79
7	120	168	121	169	122	170	123	171	124	172	125	173	126	174	127	175
8	32	80	33	81	34	82	35	83	36	84	37	85	38	86	39	87
9	128	176	129	177	130	178	131	179	132	180	133	181	134	182	135	183
10	40	88	41	89	42	90	43	91	44	92	45	93	46	94	47	95
11	136	184	137	185	138	186	139	187	140	188	141	189	142	190	143	191

Table A.10 – Bit interleaving scheme for TM3

FH cycle	f_1		f_4		f_7		f_2		f_5		f_8		f_3		f_6	
	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1
0	0	32	1	33	2	34	3	35	4	36	5	37	6	38	7	39
1	64	96	65	97	66	98	67	99	68	100	69	101	70	102	71	103
2	128	160	129	161	130	162	131	163	132	164	133	165	134	166	135	167
3	8	40	9	41	10	42	11	43	12	44	13	45	14	46	15	47
4	72	104	73	105	74	106	75	107	76	108	77	109	78	110	79	111
5	136	168	137	169	138	170	139	171	140	172	141	173	142	174	143	175
6	16	48	17	49	18	50	19	51	20	52	21	53	22	54	23	55
7	80	112	81	113	82	114	83	115	84	116	85	117	86	118	87	119
8	144	176	145	177	146	178	147	179	148	180	149	181	150	182	151	183
9	24	56	25	57	26	58	27	59	28	60	29	61	30	62	31	63
10	88	120	89	121	90	122	91	123	92	124	93	125	94	126	95	127
11	152	184	153	185	154	186	155	187	156	188	157	189	158	190	159	191

Table A.11 – Bit interleaving scheme for TM4

FH cycle	f_1		f_3		f_5		f_2		f_4	
	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1
0	0	30	60	90	120	150	180	210	1	31
1	61	91	121	151	181	211	2	32	62	92
2	122	152	182	212	3	33	63	93	123	153
3	183	213	4	34	64	94	124	154	184	214
4	5	35	65	95	125	155	185	215	6	36
5	66	96	126	156	186	216	7	37	67	97
6	127	157	187	217	8	38	68	98	128	158
7	188	218	9	39	69	99	129	159	189	219
8	10	40	70	100	130	160	190	220	11	41
9	71	101	131	161	191	221	12	42	72	102
10	132	162	192	222	13	43	73	103	133	163
11	193	223	14	44	74	104	134	164	194	224
12	15	45	75	105	135	165	195	225	16	46
13	76	106	136	166	196	226	17	47	77	107
14	137	167	197	227	18	48	78	108	138	168
15	198	228	19	49	79	109	139	169	199	229
16	20	50	80	110	140	170	200	230	21	51
17	81	111	141	171	201	231	22	52	82	112
18	142	172	202	232	23	53	83	113	143	173
19	203	233	24	54	84	114	144	174	204	234
20	25	55	85	115	145	175	205	235	26	56
21	86	116	146	176	206	236	27	57	87	117
22	147	177	207	237	28	58	88	118	148	178
23	208	238	29	59	89	119	149	179	209	239

Table A.12 – Bit interleaving scheme for TM5

FH cycle	f_1		f_4		f_7		f_2		f_5		f_8		f_3		f_6	
	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1
0	0	96	16	112	32	128	48	144	64	160	80	176	1	97	17	113
1	33	129	49	145	65	161	81	177	2	98	18	114	34	130	50	146
2	66	162	82	178	3	99	19	115	35	131	51	147	67	163	83	179
3	4	100	20	116	36	132	52	148	68	164	84	180	5	101	21	117
4	37	133	53	149	69	165	85	181	6	102	22	118	38	134	54	150
5	70	166	86	182	7	103	23	119	39	135	55	151	71	167	87	183
6	8	104	24	120	40	136	56	152	72	168	88	184	9	105	25	121
7	41	137	57	153	73	169	89	185	10	106	26	122	42	138	58	154
8	74	170	90	186	11	107	27	123	43	139	59	155	75	171	91	187
9	12	108	28	124	44	140	60	156	76	172	92	188	13	109	29	125
10	45	141	61	157	77	173	93	189	14	110	30	126	46	142	62	158
11	78	174	94	190	15	111	31	127	47	143	63	159	79	175	95	191

Table A.13 – Bit interleaving scheme for TM6

FH cycle	f_1		f_3		f_5		f_2		f_4	
	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1
0	0	120	15	135	30	150	45	165	60	180
1	75	195	90	210	105	225	1	121	16	136
2	31	151	46	166	61	181	76	196	91	211
3	106	226	2	122	17	137	32	152	47	167
4	62	182	77	197	92	212	107	227	3	123
5	18	138	33	153	48	168	63	183	78	198
6	93	213	108	228	4	124	19	139	34	154
7	49	169	64	184	79	199	94	214	109	229
8	5	125	20	140	35	155	50	170	65	185
9	80	200	95	215	110	230	6	126	21	141
10	36	156	51	171	66	186	81	201	96	216
11	111	231	7	127	22	142	37	157	52	172
12	67	187	82	202	97	217	112	232	8	128
13	23	143	38	158	53	173	68	188	83	203
14	98	218	113	233	9	129	24	144	39	159
15	54	174	69	189	84	204	99	219	114	234
16	10	130	25	145	40	160	55	175	70	190
17	85	205	100	220	115	235	11	131	26	146
18	41	161	56	176	71	191	86	206	101	221
19	116	236	12	132	27	147	42	162	57	177
20	72	192	87	207	102	222	117	237	13	133
21	28	148	43	163	58	178	73	193	88	208
22	103	223	118	238	14	134	29	149	44	164
23	59	179	74	194	89	209	104	224	119	239

Table A.14 – Bit interleaving scheme for TM7

FH cycle	f_1		f_4		f_7		f_2		f_5		f_8		f_3		f_6	
	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1
0	0	24	48	72	96	120	144	168	12	36	60	84	108	132	156	180
1	145	169	13	37	61	85	109	133	157	181	1	25	49	73	97	121
2	110	134	158	182	2	26	50	74	98	122	146	170	14	38	62	86
3	51	75	99	123	147	171	15	39	63	87	111	135	159	183	3	27
4	16	40	64	88	112	136	160	184	4	28	52	76	100	124	148	172
5	161	185	5	29	53	77	101	125	149	173	17	41	65	89	113	137
6	102	126	150	174	18	42	66	90	114	138	162	186	6	30	54	78
7	67	91	115	139	163	187	7	31	55	79	103	127	151	175	19	43
8	8	32	56	80	104	128	152	176	20	44	68	92	116	140	164	188
9	153	177	21	45	69	93	117	141	165	189	9	33	57	81	105	129
10	118	142	166	190	10	34	58	82	106	130	154	178	22	46	70	94
11	59	83	107	131	155	179	23	47	71	95	119	143	167	191	11	35

Table A.15 – Bit interleaving scheme for TM8

FH cycle	f_1		f_3		f_5		f_2		f_4	
	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1
0	0	75	15	90	30	105	45	120	60	135
1	46	121	61	136	1	76	16	91	31	106
2	17	92	32	107	47	122	62	137	2	77
3	63	138	3	78	18	93	33	108	48	123
4	34	109	49	124	64	139	4	79	19	94
5	5	80	20	95	35	110	50	125	65	140
6	51	126	66	141	6	81	21	96	36	111
7	22	97	37	112	52	127	67	142	7	82
8	68	143	8	83	23	98	38	113	53	128
9	39	114	54	129	69	144	9	84	24	99
10	10	85	25	100	40	115	55	130	70	145
11	56	131	71	146	11	86	26	101	41	116
12	27	102	42	117	57	132	72	147	12	87
13	73	148	13	88	28	103	43	118	58	133
14	44	119	59	134	74	149	14	89	29	104

Table A.16 – Bit interleaving scheme for TM9 and TM11

FH cycle	f_1		f_9		f_4		f_{12}		f_7		f_{15}		f_2		f_{10}	
	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1
0	0	16	32	48	64	80	96	112	128	144	160	176	192	208	8	24
1	65	81	97	113	129	145	161	177	193	209	9	25	41	57	73	89
2	130	146	162	178	194	210	10	26	42	58	74	90	106	122	138	154
3	195	211	11	27	43	59	75	91	107	123	139	155	171	187	203	219
4	44	60	76	92	108	124	140	156	172	188	204	220	5	21	37	53
5	109	125	141	157	173	189	205	221	6	22	38	54	70	86	102	118
6	174	190	206	222	7	23	39	55	71	87	103	119	135	151	167	183

FH cycle	f_5		f_{13}		f_8		f_{16}		f_3		f_{11}		f_6		f_{14}	
	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1
0	40	56	72	88	104	120	136	152	168	184	200	216	1	17	33	49
1	105	121	137	153	169	185	201	217	2	18	34	50	66	82	98	114
2	170	186	202	218	3	19	35	51	67	83	99	115	131	147	163	179
3	4	20	36	52	68	84	100	116	132	148	164	180	196	212	12	28
4	69	85	101	117	133	149	165	181	197	213	13	29	45	61	77	93
5	134	150	166	182	198	214	14	30	46	62	78	94	110	126	142	158
6	199	215	15	31	47	63	79	95	111	127	143	159	175	191	207	223

Table A.17 – Bit interleaving scheme for TM10

FH cycle	f_1			f_3			f_5			f_2			f_4		
	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2
0	0	75	150	15	90	165	30	105	180	45	120	195	60	135	210
1	46	121	196	61	136	211	1	76	151	16	91	166	31	106	181
2	17	92	167	32	107	182	47	122	197	62	137	212	2	77	152
3	63	138	213	3	78	153	18	93	168	33	108	183	48	123	198
4	34	109	184	49	124	199	64	139	214	4	79	154	19	94	169
5	5	80	155	20	95	170	35	110	185	50	125	200	65	140	215
6	51	126	201	66	141	216	6	81	156	21	96	171	36	111	186
7	22	97	172	37	112	187	52	127	202	67	142	217	7	82	157
8	68	143	218	8	83	158	23	98	173	38	113	188	53	128	203
9	39	114	189	54	129	204	69	144	219	9	84	159	24	99	174
10	10	85	160	25	100	175	40	115	190	55	130	205	70	145	220
11	56	131	206	71	146	221	11	86	161	26	101	176	41	116	191
12	27	102	177	42	117	192	57	132	207	72	147	222	12	87	162
13	73	148	223	13	88	163	28	103	178	43	118	193	58	133	208
14	44	119	194	59	134	209	74	149	224	14	89	164	29	104	179

Table A.18 – Bit interleaving scheme for TM12

FH cycle	f_1			f_9			f_4			f_{12}			f_7			f_{15}			f_2			f_{10}		
	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2
0	0	24	48	72	96	120	144	168	192	216	240	264	288	312	12	36	60	84	108	132	156	180	204	228
1	157	181	205	229	253	277	301	325	2	26	50	74	98	122	146	170	194	218	242	266	290	314	14	38
2	291	315	15	39	63	87	111	135	159	183	207	231	255	279	303	327	4	28	52	76	100	124	148	172
3	101	125	149	173	197	221	245	269	293	317	17	41	65	89	113	137	161	185	209	233	257	281	305	329
4	258	282	306	330	7	31	55	79	103	127	151	175	199	223	247	271	295	319	19	43	67	91	115	139
5	68	92	116	140	164	188	212	236	260	284	308	332	9	33	57	81	105	129	153	177	201	225	249	273
6	202	226	250	2747	298	322	22	46	70	94	118	142	166	190	214	238	262	286	310	334	11	35	59	83

FH cycle	f_5			f_{13}			f_8			f_{16}			f_3			f_{11}			f_6			f_{14}		
	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2
0	252	276	300	324	1	25	49	73	97	121	145	169	193	217	241	265	289	313	13	37	61	85	109	133
1	62	86	110	134	158	182	206	230	254	278	302	326	3	27	51	75	99	123	147	171	195	219	243	267
2	196	220	244	268	292	316	16	40	64	88	112	136	160	184	208	232	256	280	304	328	5	29	53	77
3	6	30	54	78	102	126	150	174	198	222	246	270	294	318	18	42	66	90	114	138	162	186	210	234
4	163	187	211	235	259	283	307	331	8	32	56	80	104	128	152	176	200	224	248	272	296	320	20	44
5	297	321	21	45	69	93	117	141	165	189	213	237	261	285	309	333	10	34	58	82	106	130	154	178
6	107	131	155	179	203	227	251	275	299	323	23	47	71	95	119	143	167	191	215	239	263	287	311	335

Table A.19 – Bit interleaving scheme for TM13 (1 of 2)

FH cycle	f_1			f_4			f_7			f_2			f_5		
	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2
0	0	18	36	54	72	90	108	126	144	162	180	198	216	234	252
1	333	351	369	387	405	423	441	459	477	9	27	45	63	81	99
2	163	181	199	217	235	253	271	289	307	325	343	361	379	397	415
3	442	460	478	10	28	46	64	82	100	118	136	154	172	190	208
4	272	290	308	326	344	362	380	398	416	434	452	470	2	20	38
5	119	137	155	173	191	209	227	245	263	281	299	317	335	353	371
6	381	399	417	435	453	471	3	21	39	57	75	93	111	129	147
7	228	246	264	282	300	318	336	354	372	390	408	426	444	462	480
8	58	76	94	112	130	148	166	184	202	220	238	256	274	292	310
9	13	31	49	67	85	103	121	139	157	175	193	211	229	247	265
10	329	347	365	383	401	419	437	455	473	5	23	41	59	77	95
11	176	194	212	230	248	266	284	302	320	338	356	374	392	410	428
12	438	456	474	6	24	42	60	78	96	114	132	150	168	186	204
13	285	303	321	339	357	375	393	411	429	447	465	483	15	33	51
14	115	133	151	169	187	205	223	241	259	277	295	313	331	349	367
15	394	412	430	448	466	484	16	34	52	70	88	106	124	142	160
16	224	242	260	278	296	314	332	350	368	386	404	422	440	458	476
17	71	89	107	125	143	161	179	197	215	233	251	269	287	305	323

Table A.19 – Bit interleaving scheme for TM13 (2 of 2)

FH cycle	f_8			f_3			f_6			f_9		
	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2
0	270	288	306	324	342	360	378	396	414	432	450	468
1	117	135	153	171	189	207	225	243	261	279	297	315
2	433	451	469	1	19	37	55	73	91	109	127	145
3	226	244	262	280	298	316	334	352	370	388	406	424
4	56	74	92	110	128	146	164	182	200	218	236	254
5	389	407	425	443	461	479	11	29	47	65	83	101
6	165	183	201	219	237	255	273	291	309	327	345	363
7	12	30	48	66	84	102	120	138	156	174	192	210
8	328	346	364	382	400	418	436	454	472	4	22	40
9	283	301	319	337	355	373	391	409	427	445	463	481
10	113	131	149	167	185	203	221	239	257	275	293	311
11	446	464	482	14	32	50	68	86	104	122	140	158
12	222	240	258	276	294	312	330	348	366	384	402	420
13	69	87	105	123	141	159	177	195	213	231	249	267
14	385	403	421	439	457	475	7	25	43	61	79	97
15	178	196	214	232	250	268	286	304	322	340	358	376
16	8	26	44	62	80	98	116	134	152	170	188	206
17	341	359	377	395	413	431	449	467	485	17	35	53

Table A.20 – Bit interleaving scheme for TM14

Symbol	f_1			f_2			f_3			f_4			f_5			f_6			f_7			f_8		
	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2
0	0	20	40	60	80	100	120	140	160	180	200	220	240	260	280	300	320	340	360	380	400	420	440	460
1	430	450	470	10	30	50	70	90	110	130	150	170	190	210	230	250	270	290	310	330	350	370	390	410
2	241	261	281	301	321	341	361	381	401	421	441	461	1	21	41	61	81	101	121	141	161	181	201	221
3	191	211	231	251	271	291	311	331	351	371	391	411	431	451	471	11	31	51	71	91	111	131	151	171
4	362	382	402	422	442	462	2	22	42	62	82	102	122	142	162	182	202	222	242	262	282	302	322	342
5	312	332	352	372	392	412	432	452	472	12	32	52	72	92	112	132	152	172	192	212	232	252	272	292
6	123	143	163	183	203	223	243	263	283	303	323	343	363	383	403	423	443	463	3	23	43	63	83	103
7	73	93	113	133	153	173	193	213	233	253	273	293	313	333	353	373	393	413	433	453	473	13	33	53
8	4	24	44	64	84	104	124	144	164	184	204	224	244	264	284	304	324	344	364	384	404	424	444	464
9	434	454	474	14	34	54	74	94	114	134	154	174	194	214	234	254	274	294	314	334	354	374	394	414
10	245	265	285	305	325	345	365	385	405	425	445	465	5	25	45	65	85	105	125	145	165	185	205	225
11	195	215	235	255	275	295	315	335	355	375	395	415	435	455	475	15	35	55	75	95	115	135	155	175
12	366	386	406	426	446	466	6	26	46	66	86	106	126	146	166	186	206	226	246	266	286	306	326	346
13	316	336	356	376	396	416	436	456	476	16	36	56	76	96	116	136	156	176	196	216	236	256	276	296
14	127	147	167	187	207	227	247	267	287	307	327	347	367	387	407	427	447	467	7	27	47	67	87	107
15	77	97	117	137	157	177	197	217	237	257	277	297	317	337	357	377	397	417	437	457	477	17	37	57
16	8	28	48	68	88	108	128	148	168	188	208	228	248	268	288	308	328	348	368	388	408	428	448	468
17	438	458	478	18	38	58	78	98	118	138	158	178	198	218	238	258	278	298	318	338	358	378	398	418
18	249	269	289	309	329	349	369	389	409	429	449	469	9	29	49	69	89	109	129	149	169	189	209	229
19	199	219	239	259	279	299	319	339	359	379	399	419	439	459	479	19	39	59	79	99	119	139	159	179

Table A.21 – Bit interleaving scheme for TM15 (1 of 2)

Symbol	f_1			f_2			f_3			f_4			f_5			f_6			f_7		
	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2
0	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200
1	481	491	501	511	521	531	541	551	561	571	581	591	1	11	21	31	41	51	61	71	81
2	362	372	382	392	402	412	422	432	442	452	462	472	482	492	502	512	522	532	542	552	562
3	243	253	263	273	283	293	303	313	323	333	343	353	363	373	383	393	403	413	423	433	443
4	124	134	144	154	164	174	184	194	204	214	224	234	244	254	264	274	284	294	304	314	324
5	5	15	25	35	45	55	65	75	85	95	105	115	125	135	145	155	165	175	185	195	205
6	486	496	506	516	526	536	546	556	566	576	586	596	6	16	26	36	46	56	66	76	86
7	367	377	387	397	407	417	427	437	447	457	467	477	487	497	507	517	527	537	547	557	567
8	248	258	268	278	288	298	308	318	328	338	348	358	368	378	388	398	408	418	428	438	448
9	129	139	149	159	169	179	189	199	209	219	229	239	249	259	269	279	289	299	309	319	329

Symbol	f_8			f_9			f_{10}			f_{11}			f_{12}			f_{13}			f_{14}		
	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2
0	210	220	230	240	250	260	270	280	290	300	310	320	330	340	350	360	370	380	390	400	410
1	91	101	111	121	131	141	151	161	171	181	191	201	211	221	231	241	251	261	271	281	291
2	572	582	592	2	12	22	32	42	52	62	72	82	92	102	112	122	132	142	152	162	172
3	453	463	473	483	493	503	513	523	533	543	553	563	573	583	593	3	13	23	33	43	53
4	334	344	354	364	374	384	394	404	414	424	434	444	454	464	474	484	494	504	514	524	534
5	215	225	235	245	255	265	275	285	295	305	315	325	335	345	355	365	375	385	395	405	415
6	96	106	116	126	136	146	156	166	176	186	196	206	216	226	236	246	256	266	276	286	296
7	577	587	597	7	17	27	37	47	57	67	77	87	97	107	117	127	137	147	157	167	177
8	458	468	478	488	498	508	518	528	538	548	558	568	578	588	598	8	18	28	38	48	58
9	339	349	359	369	379	389	399	409	419	429	439	449	459	469	479	489	499	509	519	529	539

Table A.21 – Bit interleaving scheme for TM15 (2 of 2)

Symbol	f_{15}			f_{16}			f_{17}			f_{18}			f_{19}			f_{20}		
	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2	b_0	b_1	b_2
0	420	430	440	450	460	470	480	490	500	510	520	530	540	550	560	570	580	590
1	301	311	321	331	341	351	361	371	381	391	401	411	421	431	441	451	461	471
2	182	192	202	212	222	232	242	252	262	272	282	292	302	312	322	332	342	352
3	63	73	83	93	103	113	123	133	143	153	163	173	183	193	203	213	223	233
4	544	554	564	574	584	594	4	14	24	34	44	54	64	74	84	94	104	114
5	425	435	445	455	465	475	485	495	505	515	525	535	545	555	565	575	585	595
6	306	316	326	336	346	356	366	376	386	396	406	416	426	436	446	456	466	476
7	187	197	207	217	227	237	247	257	267	277	287	297	307	317	327	337	347	357
8	68	78	88	98	108	118	128	138	148	158	168	178	188	198	208	218	228	238
9	549	559	569	579	589	599	9	19	29	39	49	59	69	79	89	99	109	119

Table A.22 – Bit interleaving scheme for TM16

FH cycle	f_1	f_4	f_7	f_2	f_5	f_8	f_3	f_6
	b_0	b_0	b_0	b_0	b_0	b_0	b_0	b_0
0	0	113	98	83	68	53	38	23
1	8	121	106	91	76	61	46	31
2	16	1	114	99	84	69	54	39
3	24	9	122	107	92	77	62	47
4	32	17	2	115	100	85	70	55
5	40	25	10	123	108	93	78	63
6	48	33	18	3	116	101	86	71
7	56	41	26	11	124	109	94	79
8	64	49	34	19	4	117	102	87
9	72	57	42	27	12	125	110	95
10	80	65	50	35	20	5	118	103
11	88	73	58	43	28	13	126	111
12	96	81	66	51	36	21	6	119
13	104	89	74	59	44	29	14	127
14	112	97	82	67	52	37	22	7
15	120	105	90	75	60	45	30	15

Table A.23 – Bit interleaving scheme for TM17

FH cycle	f_2	f_5	f_1	f_4	f_7	f_3	f_6
	b_0	b_0	b_0	b_0	b_0	b_0	b_0
0	0	85	72	59	46	33	20
1	7	92	79	66	53	40	27
2	14	1	86	73	60	47	34
3	21	8	93	80	67	54	41
4	28	15	2	87	74	61	48
5	35	22	9	94	81	68	55
6	42	29	16	3	88	75	62
7	49	36	23	10	95	82	69
8	56	43	30	17	4	89	76
9	63	50	37	24	11	96	83
10	70	57	44	31	18	5	90
11	77	64	51	38	25	12	97
12	84	71	58	45	32	19	6
13	91	78	65	52	39	26	13

Table A.24 – Bit interleaving scheme for TM18

FH cycle	f_1	f_3	f_5	f_2	f_4
	b_0	b_0	b_0	b_0	b_0
0	0	41	32	23	14
1	5	46	37	28	19
2	10	1	42	33	24
3	15	6	47	38	29
4	20	11	2	43	34
5	25	16	7	48	39
6	30	21	12	3	44
7	35	26	17	8	49
8	40	31	22	13	4
9	45	36	27	18	9

Table A.25 – Bit interleaving scheme for TM19

FH cycle	f_2		f_5		f_1		f_4		f_7		f_3		f_6	
	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1
0	0	7	169	176	142	149	115	122	88	95	61	68	34	41
1	14	21	183	190	156	163	129	136	102	109	75	82	48	55
2	28	35	1	8	170	177	143	150	116	123	89	96	62	69
3	42	49	15	22	184	191	157	164	130	137	103	110	76	83
4	56	63	29	36	2	9	171	178	144	151	117	124	90	97
5	70	77	43	50	16	23	185	192	158	165	131	138	104	111
6	84	91	57	64	30	37	3	10	172	179	145	152	118	125
7	98	105	71	78	44	51	17	24	186	193	159	166	132	139
8	112	119	85	92	58	65	31	38	4	11	173	180	146	153
9	126	133	99	106	72	79	45	52	18	25	187	194	160	167
10	140	147	113	120	86	93	59	66	32	39	5	12	174	181
11	154	161	127	134	100	107	73	80	46	53	19	26	188	195
12	168	175	141	148	114	121	87	94	60	67	33	40	6	13
13	182	189	155	162	128	135	101	108	74	81	47	54	20	27

Table A.26 – Bit interleaving scheme for TM20

FH cycle	f_1		f_3		f_5		f_2		f_4	
	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1	b_0	b_1
0	0	5	81	86	62	67	43	48	24	29
1	10	15	91	96	72	77	53	58	34	39
2	20	25	1	6	82	87	63	68	44	49
3	30	35	11	16	92	97	73	78	54	59
4	40	45	21	26	2	7	83	88	64	69
5	50	55	31	36	12	17	93	98	74	79
6	60	65	41	46	22	27	3	8	84	89
7	70	75	51	56	32	37	13	18	94	99
8	80	85	61	66	42	47	23	28	4	9
9	90	95	71	76	52	57	33	38	14	19

A.3 Frequency mapping schemes (FMS)

In the following FMS (see Table A.27 to Table A.35) $ts\varphi_n$ and φ_n symbolise PSK encoded elements assigned to the first carrier frequency within a carrier frequency assigning cycle (see 5.3.5). The symbols 0 to Nfhc-1 are the symbols within a frequency hopping cycle of Nfch symbols.

Table A.27 – Frequency mapping scheme 0 (FMS0)

Symbol	PSK encoded elements							
	$ts\varphi_n$	$ts\varphi_{n+1}$	$ts\varphi_{n+2}$	$ts\varphi_{n+3}$	$ts\varphi_{n+4}$	$ts\varphi_{n+5}$	$ts\varphi_{n+6}$	$ts\varphi_{n+7}$
0	f_3	--	--	--	--	--	--	--
1	--	f_6	--	--	--	--	--	--
2	--	--	f_1	--	--	--	--	--
3	--	--	--	f_4	--	--	--	--
4	--	--	--	--	f_7	--	--	--
5	--	--	--	--	--	f_2	--	--
6	--	--	--	--	--	--	f_5	--
7	--	--	--	--	--	--	--	f_8
A paragraph containing a requirement.								
NOTE f_1 to f_8 are carrier frequencies from CFL0								

Table A.28 – Frequency mapping scheme 1 (FMS1)

Symbol	PSK encoded elements							
	φ_n	φ_{n+1}	φ_{n+2}	φ_{n+3}	φ_{n+4}	φ_{n+5}	φ_{n+6}	φ_{n+7}
0	f_1	--	--	--	--	--	--	--
1	--	f_4	--	--	--	--	--	--
2	--	--	f_7	--	--	--	--	--
3	--	--	--	f_2	--	--	--	--
4	--	--	--	--	f_5	--	--	--
5	--	--	--	--	--	f_8	--	--
6	--	--	--	--	--	--	f_3	--
7	--	--	--	--	--	--	--	f_6

A paragraph containing a requirement.

NOTE f_1 to f_8 are carrier frequencies from CFL1

Table A.29 – Frequency mapping scheme 2 (FMS2)

Symbol	PSK encoded elements				
	φ_n	φ_{n+1}	φ_{n+2}	φ_{n+3}	φ_{n+4}
0	f_1	--	--	--	--
1	--	f_3	--	--	--
2	--	--	f_5	--	--
3	--	--	--	f_2	--
4	--	--	--	--	f_4

A paragraph containing a requirement.

NOTE f_1 to f_5 are carrier frequencies from CFL2

Table A.30 – Frequency mapping scheme 3 (FMS3)

Symbol	PSK encoded elements															
	φ_n	φ_{n+1}	φ_{n+2}	φ_{n+3}	φ_{n+4}	φ_{n+5}	φ_{n+6}	φ_{n+7}	φ_{n+8}	φ_{n+9}	φ_{n+10}	φ_{n+11}	φ_{n+12}	φ_{n+13}	φ_{n+14}	φ_{n+15}
0	f_1	f_9	--	--	--	--	--	--		--	--	--	--	--	--	--
1	--	--	f_4	f_{12}	--	--	--	--	--	--	--	--	--	--	--	--
2	--	--	--	--	f_7	f_{15}	--	--	--	--	---	--	--	--	--	--
3	--	--	--	--	--	--	f_2	f_{10}	--	--	--	--	--	--	--	--
4	--	--	--	--	-	--	--	--	f_5	f_{13}	--	--	--	--	--	--
5	--	--	--	--	--		--	--	--	--	f_8	f_{16}	--	--	--	--
6	--	--	--	--	--	--		--	--	--	--	--	f_3	f_{11}	--	--
7	--	--	--	--	--	--	--		--	--	--	--	--	--	f_6	f_{14}

A paragraph containing a requirement.

NOTE f_1 to f_{16} are carrier frequencies from CFL3

Table A.31 – Frequency mapping scheme 4 (FMS4)

Symbol	PSK encoded elements								
	φ_n	φ_{n+1}	φ_{n+2}	φ_{n+3}	φ_{n+4}	φ_{n+5}	φ_{n+6}	φ_{n+7}	φ_{n+8}
0	f_1	f_4	f_7	--	--	--	--	--	
1	--	--	--	f_2	f_5	f_8	--	--	--
2	--	--	--	--		--	f_3	f_6	f_9

A paragraph containing a requirement.

NOTE f_1 to f_9 are carrier frequencies from CFL4

Table A.32 – Frequency mapping scheme 5 (FMS5)

Symbol	PSK encoded elements							
	φ_n	φ_{n+1}	φ_{n+2}	φ_{n+3}	φ_{n+4}	φ_{n+5}	φ_{n+6}	φ_{n+7}
0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8

A paragraph containing a requirement.

NOTE f_1 to f_8 are carrier frequencies from CFL5

Table A.33 – Frequency mapping scheme 6 (FMS6)

Symbol	PSK encoded elements																			
	φ_n	φ_{n+1}	φ_{n+2}	φ_{n+3}	φ_{n+4}	φ_{n+5}	φ_{n+6}	φ_{n+7}	φ_{n+8}	φ_{n+9}	φ_{n+10}	φ_{n+11}	φ_{n+12}	φ_{n+13}	φ_{n+14}	φ_{n+15}	φ_{n+16}	φ_{n+17}	φ_{n+18}	φ_{n+19}
0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}

A paragraph containing a requirement.

NOTE f_1 to f_{20} are carrier frequencies from CFL6

Table A.34 – Frequency mapping scheme 7 (FMS7)

Symbol	PSK encoded elements						
	φ_n	φ_{n+1}	φ_{n+2}	φ_{n+3}	φ_{n+4}	φ_{n+5}	φ_{n+6}
0	f_2	--	--	--	--	--	--
1	--	f_5	--	--	--	--	--
2	--	--	f_1	--	--	--	--
3	--	--	--	f_4	--	--	--
4	--	--	--	--	f_7	--	--
5	--	--	--	--	--	f_3	--
6	--	--	--	--	--	--	f_6

A paragraph containing a requirement.

NOTE f_1 to f_7 are carrier frequencies from CFL7

Table A.35 – Frequency mapping scheme 8 (FMS8)

Symbol	PSK encoded elements				
	φ_n	φ_{n+1}	φ_{n+2}	φ_{n+3}	φ_{n+4}
0	f_1	--	--	--	--
1	--	f_3	--	--	--
2	--	--	f_5	--	--
3	--	--	--	f_2	--
4	--	--	--	--	f_4

A paragraph containing a requirement.

NOTE f_1 to f_5 are carrier frequencies from CFL8

Annex B
(normative)

Logical Link Control Functions

B.1 Master node messages for data link control (PRM=1, DLS=0)

In the fields of the messages the values are placed using the little endian system. Within an octet or field the bits with increasing number are arranged from right to left.

L_FC=0: Generate temporary link address (BROADCAST)

This message is used to initiate the generation of a temporary link address (TLA) in all unregistered slave nodes (without a link address) or in already registered slave nodes that are trying to register with a new master node. The TLA is generated as a random value in a range from 0 to a value defined by RTLA in the message. TLAs are used in the subsequent searches for new slave nodes by the master node during the registration procedure.

The message is always transmitted as a broadcast with CIN ≠ 0xFFFF and LA = 0xFFFF.

Table B.1 – Structure of the message ‘Generate temporary link address’

Byte of DB	Bit number								Additional description	
	7	6	5	4	3	2	1	0		
--	PRM =1	DLS =0	FCV =0	FCB =0	L_FC = 0				LLCF in MPDU header	
1	RTLA								[LSB]	
2									[MSB]	

The structure of the message is shown in Table B.1, where:

RTLA: range for temporary link address values. It may have any value in the range from 0x0000 to 0xFFFF.

L_FC=1: Clear transmit buffer

With this message the master node forces the addressed slave node to clear the transmit buffer of its data link control channel (DLS = 0). In this message the FCB bit is always zero, and upon receipt of these messages the data link control channel of the addressed slave node will always be set to expect the next message from the master node with DLS =0 and FCV = 1 to have an opposite setting of FCB, i. e. FCB = 1.

The structure of the message is shown in Table B.2. The message is sent with CIN ≠ 0xFFFF, LA ≠ 0xFFFF and DBL = 0.

Table B.2 – Structure of the message 'Clear transmit buffer'

<table>
<tr><th rowspan="2">Byte of DB</th><th colspan="8">Bit number</th><th rowspan="2">Additional description</th></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>--</td><td>PRM =1</td><td>DLS =0</td><td>FCV =0</td><td>FCB =0</td><td colspan="4">L_FC = 1</td><td>LLCF in MPDU header</td></tr>
</table>

L_FC=2: Master node beacon (BROADCAST)

This message is used by a master node to send a beacon to other master nodes. All master nodes (p-MN or s-MN operating as a master node) broadcast their messages periodically with a periodicity defined by the parameter MN_BEACON_P. Slave nodes and s-MNs operating as slave or master nodes evaluate this message to estimate the distance to the master node and the concentration of slave nodes in its network cell.

The message is always transmitted as a broadcast with FXS = 0xF, CIN = 0xFFFF and LA = 0xFFFF. The message may cross the boundary of the cell around the master.

Table B.3 – Structure of the message 'Master node beacon'

<table>
<tr><th rowspan="2">Byte of DB</th><th colspan="8">Bit number</th><th colspan="2" rowspan="2">Additional description</th></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>--</td><td>PRM =1</td><td>DLS =0</td><td>FCV =0</td><td>FCB =0</td><td colspan="4">L_FC = 2</td><td colspan="2">LLCF in MPDU header</td></tr>
<tr><td>1</td><td colspan="8">Reserved</td><td colspan="2"></td></tr>
<tr><td>2</td><td colspan="8">CMN</td><td colspan="2"></td></tr>
<tr><td>3</td><td colspan="8" rowspan="2">MCIN</td><td>[LSB]</td><td rowspan="2"></td></tr>
<tr><td>4</td><td>[MSB]</td></tr>
<tr><td>5</td><td colspan="8" rowspan="2">NRN</td><td>[LSB]</td><td rowspan="2"></td></tr>
<tr><td>6</td><td>[MSB]</td></tr>
<tr><td>7</td><td colspan="4">FXTT</td><td colspan="4">FXTM</td><td colspan="2"></td></tr>
<tr><td>8</td><td colspan="8" rowspan="2">NNT</td><td>[LSB]</td><td rowspan="2"></td></tr>
<tr><td>9</td><td>[MSB]</td></tr>
</table>

The structure of the message is shown in Table B.3, where:

CMN: consecutive message number enumerates with modulo 256 transmitted beacon messages. CMN may have any value in the range from 0 to 255.

MCIN: master node channel identification number contains M_CIN value of the master node transmitting the beacon message. It may have any value in the range from 0x0000 to 0xFFFE.

NRN: number of slave nodes currently registered with the master node. It may have any value in the range from 0 to a value defined by the 16 bit parameter DLLCNMAX of the master node.

FXTM: maximal a value of FXT, which is necessary for communication between the master node and slave nodes registered with this master node. FXTM = 1 means, that the master node has communicated with all of the NRN registered slave nodes using maximally 1 message retransmission (2 hops). FXTM may have any value in the range from 0 to 15.

FXTT: FXT threshold: a value of FXT, which allows communication between the master node and a number of slave nodes registered with this master node, which is contained in the field NNT. FXTT=1, NNT=20 and NRN=100 mean, for instance, that the master node needs maximally 1 message retransmission (2 hops) to communicate with 20 out of 100 slave nodes within its network cell. FXTT may have any value in the range from 0 to 15.

NNT: number of nodes under the threshold: number of slave nodes that are registered with the master node and can communicate with it, if FXT is limited by the threshold value FXTT. NNT may have any value in the range from 0 to a value defined by the 16 bit parameter DLLCNMAX of the master node.

L_FC=3: Master node deactivated (BROADCAST)

This message is used by an s-MN operating as master node to broadcast to its slave nodes, that it is going to stop operating as a master node.

The structure of the message is shown in Table B.4. The message is always transmitted as a broadcast with FXS = 0xF, CIN ≠ 0xFFFF and LA = 0xFFFF.

Table B.4 – Structure of the message ‘Master node deactivated’

Byte of DB	Bit number								Additional description
	7	6	5	4	3	2	1	0	
--	PRM =1	DLS =0	FCV =0	FCB =0	L_FC = 3				LLCF in MPDU header
1	Reserved								undefined value
2	Reserved								undefined value
3	Reserved								undefined value
4	Reserved								undefined value
5	Reserved								undefined value
6	Reserved								undefined value

L_FC=4: Clock synchronisation command (BROADCAST)

This message is used by a master node for remote clock synchronisation in slave nodes registered with this master node. The message contains the local date and time (no daylight saving time) of the master node before the beginning of the frame transmission procedure.

The structure of the message is shown in Table B.5. The message is always transmitted as a broadcast with FXS = 0xF, CIN ≠ 0xFFFF and LA = 0xFFFF.

Table B.5 – Structure of the message 'Clock synchronisation command'

Byte of DB	Bit number								Additional description	
	7	6	5	4	3	2	1	0		
--	PRM =1	DLS =0	FCV =0	FCB =0	L_FC = 4				LLCF in MPDU header	
1	Date and time in seconds [a]								[LSB]	
2										
3										
4									[MSB]	
5	Number of milliseconds past the second [b]								[LSB]	
6									[MSB]	

[a] Possible values: 0x00000000 = 1970-01-01 00:00:00 to
0xFFFFFFFF = 2106-02-07 06:28:15

[b] Valid values: 0x0000 to 0x03E7;
invalid values: 0x03E8 to 0xFFFF

L_FC=5: Reuse link address (BROADCAST)

This message is sent by the master node, if one or more slave nodes have not responded at least for a time period defined by the parameter DLLCRSTBTH or after a master node's restart from a power-down, when it still has slave nodes registered (their registration information was stored in a non-volatile memory before power-down). The message contains the local date and time (no daylight saving time) of the master node's last power-down minus 15 seconds. This message may be only sent, if the time period since the last power-down is not longer than the value defined by the parameter M_PDDTH.

The structure of the message is shown in Table B.6. The message is always transmitted as a broadcast with FXS = 0xF, CIN ≠ 0xFFFF and LA = 0xFFFF.

Table B.6 – Structure of the message 'Reuse link address'

Byte of DB	Bit number								Additional description
	7	6	5	4	3	2	1	0	
--	PRM =1	DLS =0	FCV =0	FCB =0	L_FC = 5				LLCF in MPDU header
1	Date and time in seconds [a]								[LSB]
2									
3									
4									[MSB]

[a] Possible values: 0x00000000 = 1970-01-01 00:00:00 to
0xFFFFFFFF = 2106-02-07 06:28:15

L_FC=6: Perform PHY link test

This message is sent by a master node either as unicast (to a dedicated slave node) or as a multicast/broadcast. With the unicast message the master node forces the addressed slave node to start a PHY link test by transmitting a PHY link test frame. With the multicast/broadcast message the master node signalizes to other nodes, that it is transmitting a PHY link test frame immediately afterwards.

The structure of the message is shown in Table B.7. The message is sent by the master node with CIN ≠ 0xFFFF. Depending on the value in the LA field the FCV bit and as a result the FCB bit can have different values (see the remark below).

Table B.7 – Structure of the message 'Perform PHY link test'

Byte of DB	Bit number								Additional description
	7	6	5	4	3	2	1	0	
--	PRM =1	DLS =0	FCV [a]	FCB [a]	L_FC = 6				LLCF in MPDU header
1	PLTM								

[a] If the frame is sent as a non-confirmed broadcast/multicast, the FCV and FCB bits are zero. If the frame is sent as unicast, the FCV=1 and FCB bit may be alternated by the master node.

PLTM: PHY link test mode number. The possible values of PLTM are given in Table B.8 below

Table B.8 – Values of PLTM

PLTM value	Transmission mode used for PHY link test
0 to 20	PHY link test is to be performed by the master node or the addressed slave node by transmitting the message ‘MN PHY test link’ or ‘SN PHY test link’ respectively with the TM value 0 to 20
21 to 63	Reserved for future use.
64 to 127	Reserved for user specific PHY link test modes
128 to 255	Reserved for future use

L_FC=7: MN PHY link test

This message is sent by a master node to test the PHY link to the other nodes. The message is sent with CIN ≠ 0xFFFF and LA = 0xFFFF.

This message follows the preceded ‘Perform PHY link test’ (DLS = 0, L_FC = 6) message, which is transmitted by master node as non-confirmed multicast/broadcast.

Table B.9 – Structure of the message ‘MN PHY link test’

Byte of DB	Bit number								Additional description		
	7	6	5	4	3	2	1	0			
--	PRM =1	DLS =0	FCV =0	FCB =0	L_FC = 7				LLCF in MPDU header		
1	TL										
2	DIFN										Device identifier (DID)
3	**Serial number** (eight-digit ASCII decimal number with leading zeros included)								[LSB]	**Multivendor identification number (MIN) according to DIN 43864 – 5: 2012-01**	
4											
5											
6											
7											
8											
9											
10									[MSB]		
11	**Fabrication identification** (two digit hexadecimal number in the range 0x00 to 0xFE)								[LSB]		
12									[MSB]		
13	**Manufacturer identification** (three characters from the alphabet ‘A’ to ‘Z’ The manufacturer has to obtain this identification from *FLAG Association Limited*.								[LSB]		
14											
15									[MSB]		
16	**Category** (0x0 to 0xF)										

The structure of the message is shown in Table B.9, where:

TL: transmit level. It contains a value in dB/mV of the signal transmitted by the master node. The TL values are in unsigned 7.1-fix-point-twos-complement presentation in the range from 0x00 to 0xFF.

DIFN: device identifier format number. DIFN contains the value zero defining subsequent 14 bytes of the multivendor identification number (MIN) according to

DIN 43863–5: 2012-04 as the base for the DID. The DIFN values 1 to 255 are reserved for future use.

L_FC=8: Request to node with temporary link address (BROADCAST)

This message is used by a master node to search for one or more not yet registered slave nodes, that have generated a temporary link address (TLA) after a prior receipt of the master node message ‘Generate temporary link address’ (PRM = 1, DLS = 0, L_FC = 0).

This message is sent by a master node as a broadcast with FXS = 0xF, CIN ≠ 0xFFFF and LA = 0xFFFF during a registration procedure. The master runs the registration procedure periodically in background.

Table B.10 – Structure of the message ‘Request to node with temporary link address’

Byte of DB	Bit number								Additional description	
	7	6	5	4	3	2	1	0		
--	PRM =1	DLS =0	FCV =0	FCB =0	L_FC = 8				LLCF in MPDU header	
1	TLA								[LSB]	
2									[MSB]	

The structure of the message is shown in Table B.10, where:

TLA: temporary link address for search for slave node(s). It may have any value in the range from 0x0000 to 0xFFFF.

L_FC=9: Assign new address

This message is used by a master node to assign a new S_CIN, a new link address (S_LA) and, if necessary, one or more multicast addresses (S_MAx) to the slave node with a dedicated Device Identifier (DID) during the registration procedure or to assign new multicast addresses. The message is sent by master node with FXS = 0xF, and CIN ≠ 0xFFFF. Depending on the value in the LA field the FCV bit and as a result the FCB bit can have different values (see a remark below).

Table B.11 – Structure of the message 'Assign new address'

Byte of DB	Bit number								Additional description		
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**			
--	PRM =1	DLS =0	FCV[a]	FCB[a]	L_FC = 9				LLCF in MPDU header		
1	DIFN										Device identifier (DID)
2	**Serial number** (eight-digit ASCII decimal number with leading zeros included)								[LSB]	**Multivendor identification number (MIN) according to DIN 43863 – 5: 2012-04**	
3											
4											
5											
6											
7											
8											
9									[MSB]		
10	**Fabrication identification** (two digit hexadecimal number in the range 0x00 to 0xFE)								[LSB]		
11									[MSB]		
12	**Manufacturer identification** (three characters from the alphabet 'A' to 'Z' The manufacturer has to obtain this identification from *FLAG Association Limited*.								[LSB]		
13											
14									[MSB]		
15	**Category** (0x0 to 0xF)										
16	NLA								[LSB]		
17									[MSB]		
18	NNMA										
19	NMA1[b]								[LSB]		
20									[MSB]		
21 to 16+ 2x	NMA2 to NMAx-1[b]								[LSB] [MSB]		
17+ 2x	NMA_x[b]								[LSB]		
18+ 2x									[MSB]		

[a] The FCV and FCB bits are zero, if LA=0xFFFF. If LA≠0xFFFF, the FCV=1 and FCB bit may be alternated by the master node.

[b] The fields NMA1 to NMAx are not contained in the message if NNMA = 0.

The structure of the message is shown in Table B.11, where:

DIFN: device identifier format number. DIFN contains the value zero defining subsequent 14 byte of the multivendor identification number (MIN) according to DIN 43863–5: 2012-04 as the base for DID. The DIFN values 1 to 255 are reserved for future use.

NLA: new link address of the slave node used for MAC layer addressing. NLA may have any value in the range from 0x0000 to 0xFFFD.

NNMA: number of new multicast addresses in the message. NNMA may have any value in the range from 0 to 31. The NNMA values 32 to 118 are reserved. The NNMA values 119 to 255 are invalid. There are no new multicast addresses in the message if NNMA = 0.

NMA1 to NMAx: new multicast address 1 to x, x = NNMA for MAC layer multicast addressing. Each of the NMAx may have any value in the range from 0x0000 to 0xFFFD.

L_FC=10: Status enquiry

This message is used by the master node during the registration procedure to optimize the frame transmission parameters.

The structure of the message is shown in Table B.12. The message is sent by the master node with CIN ≠ 0xFFFF, LA ≠ 0xFFFF and DBL = 0.

Table B.12 – Structure of the message 'Status enquiry'

Byte of DB	Bit number								Additional description
	7	6	5	4	3	2	1	0	
--	PRM =1	DLS =0	FCV =1	FCB	L_FC = 10				LLCF in MPDU header

L_FC=11: Node enquiry

With this message the master node continuously polls the registered slave nodes.

The structure of the message is shown in Table B.13. The message is sent by the master node with CIN ≠ 0xFFFF, LA ≠ 0xFFFF and DBL = 0.

Table B.13 – Structure of the message 'Node enquiry'

Byte of DB	Bit number								Additional description
	7	6	5	4	3	2	1	0	
--	PRM =1	DLS =0	FCV =1	FCB	L_FC = 11				LLCF in MPDU header

L_FC=12: Data availability enquiry (BROADCAST)

This message is used by a master node to periodically initiate a non-contention free quick-check procedure to indentify data availability in the slave nodes.

The message is sent by master node as a broadcast with FXS = 0xF, CIN ≠ 0xFFFF and LA = 0xFFFF.

Table B.14 – Structure of the message 'Data availability enquiry'

Byte of DB	Bit number								Additional description	
	7	6	5	4	3	2	1	0		
--	PRM =1	DLS =0	FCV =0	FCB =0	L_FC = 12				LLCF in MPDU header	
1	RTM	TNCW								
2	CWD								[LSB]	
3									[MSB]	

The structure of the message is shown in Table B.14, where:

TNCW: Total number of contention windows in a series following the message. The slave node, which has data ready to transmit to the master node, uses for its response one of the windows in a way depending on the TNCW, which may have following values:

- 0: only one contention window is available. The start of the response inside the window shall be randomly chosen by the slave node.
- 1: only one contention window is available. The slave node shall transmit it response at the beginning of the window;
- 2 to 127: contention windows are available. The slave node randomly chooses one of the contention windows and responds at its beginning.

RTM: Reuse transmit mode (TM) from the message. RTM may have either one of the following two values:

0: TM of the response may be defined by slave node.
1: Slave node shall reuse the TM, contained in the PHY-header, to respond.

CWD Duration of a contention window in milliseconds with following values:

0 to 49: unused values;
50 to 65535: valid duration values in milliseconds.

L_FC=15: Link quality enquiry

This message is used by the master node to test the quality of the multihop link between the master node and an addressed slave node. It is sent by the master node to activate internal test functions.

The structure of the message is shown in Table B.15. The message is sent by the master node with CIN ≠ 0xFFFF and LA ≠ 0xFFFF.

Table B.15 – Structure of the message 'Link quality enquiry'

Byte of DB	Bit number								Additional description	
	7	6	5	4	3	2	1	0		
--	PRM =1	DLS =0	FCV =1	FCB	L_FC = 15				LLCF in MPDU header	
1 to 40	Random bit array (40 Bytes)								[LSB] ... [MSB]	Non-specified data

B.2 Master node messages for higher layer servicing (PRM=1, DLS=1)

In the fields of the messages the values are placed using the little endian system. Within an octet or field the bits with increasing number are arranged from right to left.

The first byte of the data block in the messages, defined in this section, contains the following two 4 bit fields:

DP: data priority, which is used during the transmission of the higher layer PDUs to one or more slave nodes. DP may have one of the following values:
- 0: lowest priority;
- 1 to 14: priority 1 to 14;
- 15: highest priority.

LCN: link channel number. LCN may have one of the following values:
- 0 to 4 and 8 to 15: reserved for higher layer PDUs;
- 5: reserved for IPv4-PDUs;
- 6: reserved for IPv6-PDUs;
- 7: reserved for DLMS/COSEM-PDUs.

For each of the link channels the master node uses a separate FCB-Bit in addition to the FCB-Bit for the data link control channel (DLS=0).

L_FC=1: Clear transmit buffer

With this message the master node forces the addressed slave node to delete the content of the transmit buffer, which corresponds to the link channel indicated by LCN. In this message the FCB bit is always zero, and upon receipt of these messages the addressed link channel of the slave node will always be set to expect the next message from the master node with FCV = 1, DLS =1 and the same LCN value to have an opposite setting of FCB, i. e. FCB = 1.

The structure of the message is shown in Table B.16. The message is sent by the master node with CIN ≠ 0xFFFF, LA ≠ 0xFFFF and DBL = 1.

Table B.16 – Structure of the message 'Clear transmit buffer'

Byte of DB	Bit number								Additional description
	7	6	5	4	3	2	1	0	
--	PRM =1	DLS =1	FCV =0	FCB =0	L_FC = 1				LLCF in MPDU header
1	DP				LCN				

L_FC=3: Data transmission

The master node uses this message to send a higher layer PDU over a specific link channel to the corresponding higher layer entity in the addressed slave node.

The message is sent by the master node with CIN ≠ 0xFFFF and LA ≠ 0xFFFF.

Table B.17 – Structure of the message 'Data transmission'

Byte of DB	Bit number								Additional description	
	7	6	5	4	3	2	1	0		
--	PRM =1	DLS =1	FCV =1	FCB	L_FC = 3				LLCF in MPDU header	
1	DP				LCN					
2	MCN									
3	Higher-layer PDU								[LSB]	
to										
n [a]									[MSB]	

[a] The maximum value of DBL = n in bytes of frames with DLS = 1 is defined by the parameter DBLMAX.

The structure of the message is shown in Table B.17, where:

MCN: multicast number. It corresponds to the *DataIdentifier*, which is used upon the invocation of a LLC layer services by a higher layer entity. MCN identifies a buffer for data storage, which is used during the data transmission procedure. The maximum possible number of buffers for all LCNs is 200. MCSN may have the following values:

- 0 to 199: informs the addressed slave node, that the higher layer PDU has been repeated because of a missing acknowledgement to the PDU transmission from the buffers 0 to 199;
- 200 to 254: reserved;
- 255: the number identifies an unicast transmission to be acknowledged by addressed slave node.

L_FC=4: Data multicast (MULTICAST/BROADCAST)

The master node uses this message to send a higher layer PDU as a broadcast or a multicast transmission over a specific link channel to the corresponding higher layer entities of one or more slave nodes.

The message is sent by the master node with CIN ≠ 0xFFFF.

Table B.18 – Structure of the message 'Data multicast'

Byte of DB	Bit number								Additional description
	7	6	5	4	3	2	1	0	
--	PRM =1	DLS =1	FCV =0	FCB =0	L_FC = 4				LLCF in MPDU header
1	DP				LCN				
2	MCN								
3 to n [a]	Higher-layer PDU								[LSB] … [MSB]

[a] The maximum value of DBL = n in bytes of frames with DLS = 1 is defined by the parameter DBLMAX.

The structure of the message is shown in Table B.18, where:

MCN: multicast number. It corresponds to the *DataIdentifier*, which is used upon the invocation of an LLC layer services by a higher layer entity. MCN identifies a buffer for data storage, which is used during the data transmission procedure. The maximum possible number of buffers for all LCNs is 200. MCSN may have the following values:

0 to 199: number identifies multicast or broadcast transmissions from the buffer 0 to 199 respectively, which is to be acknowledged by slave nodes by transmission of separate data acknowledgments over the data link control channel (DLS=0) during polling;

200 to 253, 255: reserved;

254: the number identifies a transmission, which has not to be acknowledged.

L_FC=5: Segmented data transmission

The master node uses this message to send a segmented higher layer PDU over a specific link channel to the corresponding higher layer entity in the addressed slave node. This message supports transmission of up to 16 segments of a segmented higher layer PDU with the maximum total length equal 16 x (DBLMAX – 3) bytes. The segments of the higher layer PDU shall be transmitted in an ascending sequence of messages, starting with the segment number 0.

The message is sent by the master node with CIN ≠ 0xFFFF and LA ≠ 0xFFFF.

Table B.19 – Structure of the message 'Segmented data transmission'

<table>
<tr><th rowspan="2">Byte of DB</th><th colspan="8">Bit number</th><th colspan="2" rowspan="2">Additional description</th></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>--</td><td>PRM =1</td><td>DLS =1</td><td>FCV =1</td><td>FCB</td><td colspan="4">L_FC = 5</td><td colspan="2">LLCF in MPDU header</td></tr>
<tr><td>1</td><td colspan="4">DP</td><td colspan="4">LCN</td><td colspan="2" rowspan="3"></td></tr>
<tr><td>2</td><td colspan="8">MCN</td></tr>
<tr><td>3</td><td colspan="4">MNS</td><td colspan="4">CSN</td></tr>
<tr><td rowspan="3">4
to
n [a]</td><td colspan="8" rowspan="3">Higher-layer PDU segment</td><td>[LSB]</td><td rowspan="3"></td></tr>
<tr><td></td></tr>
<tr><td>[MSB]</td></tr>
<tr><td colspan="11">[a] The maximum value of DBL = n in bytes of frames with DLS = 1 is defined by the parameter DBLMAX.</td></tr>
</table>

The structure of the message is shown in Table B.19, where:

MCN: multicast number. It corresponds to the *DataIdentifier*, which is used upon the invocation of a LLC layer services by a higher layer entity. MCN identifies the buffer for data storage, which is used during the data transmission procedure. MCN may only have the value 255, which identifies a unicast transmission to be acknowledged by the addressed slave node. Other values of MCN are reserved.

MNS: Maximal number of data segment corresponding to the total number of data segments reduced by one. It has the same value in all messages until the completion of transmission of all data segments. MNS may have following values:

- 0: unused value;
- 1 to 15: possible values.

CSN: Current number of the data segment. It may have any values in the range 0 to 15.

L_FC=6: Segmented data multicast (MULTICAST/BROADCAST)

The master node uses this message to send a segmented higher layer PDU as broadcast or a multicast transmission, which does not require acknowledgements, over a specific link channel to the corresponding higher layer entities of one or more slave nodes. This message supports transmission of up to 16 segments of a segmented higher layer PDU with the maximum total length equal 16 x (DBLMAX – 3) bytes. The segments of the higher layer PDU shall be transmitted in an ascending sequence of messages, starting with the segment number 0.

The message is sent by the master node with CIN ≠ 0xFFFF.

Table B.20 – Structure of the message ‘Segmented data multicast’

Byte of DB	Bit number								Additional description	
	7	6	5	4	3	2	1	0		
--	PRM =1	DLS =1	FCV =0	FCB =0	L_FC = 6				LLCF in MPDU header	
1	DP				LCN					
2	MCN									
3	MNS				CSN					
4 to n [a]	Higher-layer PDU segment								[LSB] [MSB]	

[a] The maximum value of DBL = n in bytes of frames with DLS = 1 is defined by the parameter DBLMAX.

The structure of the message is shown in Table B.20, where:

MCN: multicast number. It corresponds to the *DataIdentifier*, which is used upon the invocation of an LLC layer service by a higher layer entity. MCN identifies the buffer for data storage, which is used during the data transmission procedure. MCN may only have the value 254 identifying transmission, which has not to be acknowledged. Other values of MCN are reserved.

MNS: Maximal number of data segments corresponding to the total number of data segments reduced by one. It has the same value in all messages until the completion of the transmission of all data segments. MNS may have the following values:

- 0: unused value;
- 1 to 15: possible values.

CSN: Current number of the data segment. It may have any value in the range 0 to 15.

L_FC=11: Data enquiry

This message is used by the master node to request the data of a dedicated link channel from the addressed slave node.

The structure of the message is shown in Table B.21. The message is sent by the master node with CIN ≠ 0xFFFF, LA ≠ 0xFFFF and DBL = 1.

Table B.21 – Structure of the message 'Data enquiry'

Byte of DB	Bit number								Additional description
	7	6	5	4	3	2	1	0	
--	PRM =1	DLS =1	FCV =1	FCB	L_FC = 11				LLCF in MPDU header
1	DP				LCN				

B.3 Slave node messages for data link control functions (PRM=0, DLS=0)

In the fields of the messages the values are placed using the little endian system. Within an octet or field the bits with increasing number are arranged from right to left.

L_FC=0: ACK

This message is sent by a slave node as response to a SEND service expecting a CONFIRM.

The structure of the message is shown in Table B.22. The message is sent with CIN ≠ 0xFFFF, LA ≠ 0xFFFF and DBL = 0.

Table B.22 – Structure of the message ‘ACK’

Byte of DB	Bit number								Additional description
	7	6	5	4	3	2	1	0	
--	PRM =0	DLS =0	RES =0	DFC	L_FC = 0				LLCF in MPDU header

L_FC=7: SN PHY link test

This message is sent by a slave node to test the PHY links to the other nodes. The sending of this message is initiated by a preceded reception of the 'Perform PHY link test' (DLS = 0, L_FC = 6), which has been received as unicast frame from the master node, the slave node is registered with. The message is sent with CIN ≠ 0xFFFF and LA = 0xFFFF.

Table B.23 – Structure of the message 'SN PHY link test'

<table>
<tr><th rowspan="2">Byte of DB</th><th colspan="8">Bit number</th><th colspan="3" rowspan="2">Additional description</th></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>--</td><td>PRM =0</td><td>DLS =0</td><td>RES =0</td><td>DFC</td><td colspan="4">L_FC = 7</td><td colspan="3">LLCF in MPDU header</td></tr>
<tr><td>1</td><td colspan="8">TL</td><td colspan="3"></td></tr>
<tr><td>2</td><td colspan="8">DIFN</td><td colspan="2"></td><td rowspan="15">Device identifier (DID)</td></tr>
<tr><td>3</td><td colspan="8" rowspan="8">Serial number
(eight-digit ASCII decimal number
with leading zeros included)</td><td>[LSB]</td><td rowspan="14">Multivendor identification number (MIN) according to DIN 43864 – 5: 2012-01</td></tr>
<tr><td>4</td><td></td></tr>
<tr><td>5</td><td></td></tr>
<tr><td>6</td><td></td></tr>
<tr><td>7</td><td></td></tr>
<tr><td>8</td><td></td></tr>
<tr><td>9</td><td></td></tr>
<tr><td>10</td><td>[MSB]</td></tr>
<tr><td>11</td><td colspan="8" rowspan="2">Fabrication identification
(two digit hexadecimal number in the range 0x00 to 0xFE)</td><td>[LSB]</td></tr>
<tr><td>12</td><td>[MSB]</td></tr>
<tr><td>13</td><td colspan="8" rowspan="3">Manufacturer identification
(three characters from the alphabet 'A' to 'Z'
The manufacturer has to obtain this identification from FLAG Association Limited.</td><td>[LSB]</td></tr>
<tr><td>14</td><td></td></tr>
<tr><td>15</td><td>[MSB]</td></tr>
<tr><td>16</td><td colspan="8">Category (0x0 to 0xF)</td><td></td></tr>
</table>

The structure of the message is shown in Table B.23, where:

TL: transmit level. It contains a value in dB/mV of the signal transmitted by the slave node. The TL values are in unsigned 7.1-fix-point-twos-complement presentation in the range from 0x00 to 0xFF.

DIFN: device identifier format number. DIFN contains the value zero defining subsequent 14 byte of the multivendor identification number (MIN) according to DIN 43863–5: 2012-04 as the base for DID. The DIFN values 1 to 255 are reserved for future use.

L_FC=8: Status response

This message is sent by the slave node as a response to ‘Status enquiry’ (DLS = 0, L_FC = 10) from the master node. The message is sent with CIN ≠ 0xFFFF and LA ≠ 0xFFFF.

Table B.24 – Structure of the message ‘Status response’

<table>
<tr><th rowspan="2">Byte of DB</th><th colspan="8">Bit number</th><th colspan="3" rowspan="2">Additional description</th></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>--</td><td>PRM =0</td><td>DLS =0</td><td>RES =0</td><td>DFC</td><td colspan="4">L_FC = 8</td><td colspan="3">LLCF in MPDU header</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td colspan="2">RID</td><td colspan="4">HPN</td><td colspan="3"></td></tr>
<tr><td>2</td><td colspan="8">DIFN</td><td colspan="2"></td><td rowspan="15">Device identifier (DID)</td></tr>
<tr><td>3</td><td colspan="8" rowspan="8">Serial number
(eight-digit ASCII decimal number with leading zeros included)</td><td>[LSB]</td><td rowspan="14">Multivendor identification number (MIN) according to DIN 43863–5:2012-04</td></tr>
<tr><td>4</td><td></td></tr>
<tr><td>5</td><td></td></tr>
<tr><td>6</td><td></td></tr>
<tr><td>7</td><td></td></tr>
<tr><td>8</td><td></td></tr>
<tr><td>9</td><td></td></tr>
<tr><td>10</td><td>[MSB]</td></tr>
<tr><td>11</td><td colspan="8" rowspan="2">Fabrication identification
(two digit hexadecimal number in the range 0x00 to 0xFE)</td><td>[LSB]</td></tr>
<tr><td>12</td><td>[MSB]</td></tr>
<tr><td>13</td><td colspan="8" rowspan="3">Manufacturer identification
(three characters from the alphabet ‘A’ to ‘Z’
The manufacturer has to obtain this identification from FLAG Association Limited.</td><td>[LSB]</td></tr>
<tr><td>14</td><td></td></tr>
<tr><td>15</td><td>[MSB]</td></tr>
<tr><td>16</td><td colspan="8">Category (0x0 to 0xF)</td><td></td></tr>
<tr><td>17</td><td>LC7</td><td>LC6</td><td>LC5</td><td>LC4</td><td>LC3</td><td>LC2</td><td>LC1</td><td>LC0</td><td colspan="3" rowspan="2">Link channel usage flags</td></tr>
<tr><td>18</td><td>LC15</td><td>LC14</td><td>LC13</td><td>LC12</td><td>LC11</td><td>LC10</td><td>LC9</td><td>LC8</td></tr>
<tr><td>19 to 53</td><td colspan="8">Manufacturer specific ASCII data(unused data bytes are filled with 0x00)</td><td colspan="3"></td></tr>
</table>

The structure of the message is shown in Table B.24, where:

HPN: Hop plane number. It indicates the value of FXDC in the 1st successfully received copy of the preceded ‘Status inquiry’ message. HPN may contain any value in the range from 0 to 15. The maximal value of HPN may be limited by parameter FXTMAX.

RID: receive input identifier. It indicates the input grid connectors, from which the preceded ‘Status inquiry’ message was successfully received. RID may contain following values:

0: not used;
1: reception between L1 and N or L1 and L2;
2: reception between L2 and N or L2 and L3;
3: reception between L3 and N or L3 and L1;

DIFN: device identifier format number. DIFN contains the value zero defining subsequent 14 byte of the multivendor identification number (MIN) according to DIN 43863–5:2012-04 as the base for DID. The DIFN values 1 to 255 are reserved for future use.

LC0 to LC15: flag bits of the link channel map indicating the usage of the data link channels 0 to 15 respectively by the slave node. Multiple bits LCx, x=0 to 15 may have either one of the following two values:
- 1: link channel is used by the slave node;
- 0: link channel is not used by the slave node.

L_FC=9: Node response

This message is sent by a slave node to respond to a ‘Node enquiry’ (DLS = 0, L_FC = 11) from the master node. The message is sent with CIN ≠ 0xFFFF and LA ≠ 0xFFFF.

Table B.25 – Structure of the message ‘Node response’

Byte of DB	Bit number								Additional description
	7	6	5	4	3	2	1	0	
--	PRM =0	DLS =0	RES =0	DFC	L_FC = 9				LLCF in MPDU header
1 [a]	DP				LCN				Indication of higher priority data
2 [a]	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0	Data indication flag bits
3 [a]	DR15	DR14	DR13	DR12	DR11	DR10	DR9	DR8	
4 [a]	MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0	Multicast/broadcast confirmation flag bits
5 [a]	MC15	MC14	MC13	MC12	MC11	MC10	MC9	MC8	
6 to 27 [a]	MC15 to MC191								
28 [a]	MC 199	MC 198	MC 197	MC 196	MC 195	MC 194	MC 193	MC 192	

[a] The data block length (DBL) can vary between 0 and 28 bytes. If a slave node does neither have data ready for transmission, nor has set a confirmation flag, then the message is sent without a data-block (DBL = 0). If only one link channel of the slave node has data for transmission and there are no multicast/broadcast confirmations, then the data-block (DB) of the message contains only the 1st byte with data priority (DP) and the link channel number (LCN). If at least two link channels of the slave node have data for transmission or there is at least one multicast/broadcast confirmation, then the first n bytes of the data block (DB) with the indication of higher priority data and/or set flag bits are transmitted in the message, where the byte number n (n = 1 to 28) is the last byte of the DB with at least one set bit. If the slave node sends a message only with set multicast/ broadcast confirmation flag bits, then the first three bytes of the DB contain all zeros.

The structure of the message is shown in Table B.25, where:

LCN: Number of link channel containing data with higher priority ready for transmission. LCN may have one of the values in the range 0 to 15.

DP: Priority of data with corresponding LCN. It may have one of the following values:
- 0: lowest priority;
- 1 to 14: priority values 1 to 14;
- 15: highest priority.

DR0 to DR15: flag bits indicating data ready for transmission over the link channel 0 to 15 respectively. Multiple bits DRx, x=0 to 15 may have either one of the following two values:
- 1: data ready for transmission over link channel x is available;
- 0: no data for transmission over link channel x is available.

MC0 to MC199: flag bits confirming the successful reception of data transmitted by last multicast or broadcast transmission number (MCN) 0 to 199 respectively. Multiple bits MCk, k=0 to 199 may have either one of the following two values:

1: confirmation for data reception;
0: data reception is has been not confirmed.

L_FC=10: Response from node with temporary link address

This message is sent by the slave node as a response to ‘Request to node with temporary link address’ (DLS = 0, L_FC = 8) from the master node. It allows the master node to identify the responding slave node unambiguously from amongst other slave nodes in the network. The message is sent with CIN ≠ 0xFFFF and LA = 0xFFFF.

Table B.26 – Structure of the message ‘Response from node with temporary link address’

<table>
<tr><th rowspan="2">Byte of DB</th><th colspan="8">Bit number</th><th colspan="3" rowspan="2">Additional description</th></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>--</td><td>PRM =0</td><td>DLS =0</td><td>RES =0</td><td>DFC</td><td colspan="4">L_FC = 10</td><td colspan="3">LLCF in MPDU header</td></tr>
<tr><td>1</td><td colspan="2">RRI</td><td colspan="2">RID</td><td colspan="4">HPN</td><td colspan="3"></td></tr>
<tr><td>2</td><td colspan="8">DIFN</td><td colspan="2"></td><td rowspan="15">Device identifier (DID)</td></tr>
<tr><td>3</td><td colspan="8" rowspan="8">Serial number
(eight-digit ASCII decimal number with leading zeros included)</td><td>[LSB]</td><td rowspan="14">Multivendor identification number (MIN) according to DIN 43863 – 5: 2012-04</td></tr>
<tr><td>4</td><td></td></tr>
<tr><td>5</td><td></td></tr>
<tr><td>6</td><td></td></tr>
<tr><td>7</td><td></td></tr>
<tr><td>8</td><td></td></tr>
<tr><td>9</td><td></td></tr>
<tr><td>10</td><td>[MSB]</td></tr>
<tr><td>11</td><td colspan="8" rowspan="2">Fabrication identification
(two digit hexadecimal number in the range 0x00 to 0xFE)</td><td>[LSB]</td></tr>
<tr><td>12</td><td>[MSB]</td></tr>
<tr><td>13</td><td colspan="8" rowspan="3">Manufacturer identification
(three characters from the alphabet ‘A’ to ‘Z’
The manufacturer has to obtain this identification from FLAG Association Limited.</td><td>[LSB]</td></tr>
<tr><td>14</td><td></td></tr>
<tr><td>15</td><td>[MSB]</td></tr>
<tr><td>16</td><td colspan="8">Category (0x0 to 0xF)</td><td></td></tr>
<tr><td>17</td><td>LC7</td><td>LC6</td><td>LC5</td><td>LC4</td><td>LC3</td><td>LC2</td><td>LC1</td><td>LC0</td><td colspan="3" rowspan="2">Link channel usage flags</td></tr>
<tr><td>18</td><td>LC15</td><td>LC14</td><td>LC13</td><td>LC12</td><td>LC11</td><td>LC10</td><td>LC9</td><td>LC8</td></tr>
</table>

The structure of the message is shown in Table B.26, where:

HPN: Hop plane number. It indicates the value of FXDC in the 1st successfully received copy of the preceded ‘Status inquiry’ message. HPN may contain any value in the range from 0 to 15. The maximal value of HPN may be limited by parameter FXTMAX.

RID: receive input identifier. It indicates the input grid connectors, from which the preceded ‘Status inquiry’ message was successfully received. RID may contain the following values:

0: not used;
1: reception between L1 – N or L1 – L2;
2: reception between L2 – N or L2 – L3;
3: reception between L3 – N or L3 – L1;

RRI: registration reason identifier. It indicates the reason, why the slave node tries to register with the master node. RRI may contain the following values:

0: new registration. The slave node does not have any link address (S_LA = 0xFFFF);
1: re-registration with a different master node. The slave node is already registered with a master node but tries to re-register with the different master node, it is responding to;
2: re-registration after forced deregistration. The slave node tries to register on the master node, it is responding to, because the previous master node forced it to deregister by broadcasting 'Master node deactivated' (DLS = 0, L_FC = 3);
3: reserved;

DIFN: device identifier format number. DIFN contains the value zero defining subsequent 14 byte of the multivendor identification number (MIN) according to DIN 43863–5:2012-04 as the base for DID. The DIFN values 1 to 255 are reserved for future use.

LC0 to LC15: flag bits of the link channel map indicating the using of the data link channels 0 to 15 respectively by the slave node. Multiple bits LCx, x=0 to 15 may have either one of the following two values:

1: link channel is used by the slave node;
0: link channel is not used by the slave node.

L_FC=12: Data availability status

This message is sent by a node to respond to the ‘Data availability enquiry’ (DLS = 0, L_FC = 12) from the master node. The message is sent with CIN ≠ 0xFFFF and LA ≠ 0xFFFF. If the slave node does not have any data ready for transmission, the message is not sent.

Table B.27 – Structure of the message ‘Data availability status’

Byte of DB	**Bit number**								**Additional description**
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
--	PRM =0	DLS =0	RES =0	DFC	L_FC = 12				LLCF in MPDU header
1 [a]	DP				LCN				Indication of higher priority data
2 [a]	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0	Data indication flag bits
3 [a]	DR15	DR14	DR13	DR12	DR11	DR10	DR9	DR8	

[a] The length of the Data Block (DB) can vary between 1 and 3 bytes depending on the availability of data in the slave node. If, only one link channel of slave node has data, which are ready for transmission, then the DB of the message contains only the 1st byte with the data priority (DP) and the link channel number (LCN). If two or more link channels contain data to transmit, the 1st byte of DB contains the DP and the LCN of the data with a higher priority. The further one or two bytes indicating availability of data for transmission are contained in the DB, depending on the data availability in the link channels number 0 to 7 or 8 to 15 respectively.

The structure of the message is shown in Table B.27, where:

LCN: Number of link channel containing data with higher priority ready for transmission. LCN may have one of the values in the range 0 to 15.

DP: Priority of data with corresponding LCN. It may have one of the following values:
- 0: lowest priority;
- 1 to 14: priority values 1 to 14;
- 15: highest priority.

DR0 to DR15: flag bits indicating data ready for transmission over the link channel 0 to 15 respectively. Multiple bits DRx, x=0 to 15 may have either one of the following two values:
- 1: data ready for transmission over link channel x are available;
- 0: no data for transmission over link channel x are available.

L_FC=15: Link quality response

This message is sent from the slave node as a response to 'Link quality enquiry' (DLS = 0, L_FC = 12). It contains PHY quality data sets collected by the slave node upon reception of each enquiry message copy with a corresponding FXDC value during the transmission procedure from the master node. The length of the Data Block (DB) can vary depending on the number of received enquiry copies. The message is sent by the master node with CIN ≠ 0xFFFF, LA ≠ 0xFFFF. If the slave node did not receive any copies of the preceded enquiry, the message is not sent.

Table B.28 – Structure of the message 'Link quality response'

Byte of DB	Bit number								Additional description
	7	6	5	4	3	2	1	0	
--	PRM =0	DLS =0	RES =0	DFC	L_FC = 15				LLCF in MPDU header
1	NQDS								
2	RL0								PHY quality data set 0
3	SNR0								
4	BER0								
5 to 3N-2 [a]	RL1, SNR1, BER1 to RLx, SNRx, BERx								PHY quality data set 1 to N-2 [a]
3N-1 [a]	RLx								PHY quality data set N-1 [a]
3N [a]	SNRx								
3N+1 [a]	BERx								

[a] N = NQDS. N may have any value in the range from 1 to 16.

The structure of the message is shown in Table B.28, where:

NQDS: Number of PHY quality data sets (NQSN = N, N = 1 to 16). It indicates the number of the PHY quality data sets contained in the message. NQDS may have any value in the range from 1 to 16. The values 0 and 17 to255 are unused.

RL0 to RLx, x = 0 to N-1: estimated receive level. It contains a receive level estimation in dB(mV) for the message copy with FXDC = x = 0 to N-1 respectively. It contains a value in signed 8.0-fix-point-twos-complement presentation. RLx, x = 0 to N-1 may have one of the have following values:

0x00 to 0x7F and 0x81 to 0xFF: possible values;
0x80: no message was received.

SNR0 to SNRx, x = 0 to N-1: estimated receive level. It contains a receive level estimation in dB for the message copy with FXDC = x = 0 to N-1 respectively. It contains a value in signed 8.0-fix-point-twos-complement presentation. SNRx, x = 0 to N-1 may have one of the have following values:

0x00 to 0x7F and 0x81 to 0xFF: possible values;
0x80: no message was received.

BER0 to BERx, x = 0 to N-1: estimated bit error rate. It contains a result of the following expression in unsigned 8.0-fix-point-twos-complement presentation: $BERx = -50Log_{10}(ERx)$, ERx, x = 0 to N-1 is an estimation of the bit error rate of the encoded bit sequence before data bit decision of the message copy with FXDC = x = 0 to N-1 respectively. BERx, x = 0 to

N-1 may have one of the have following values:
0x00: no message was received.
0x01 to 0xFF: possible values.

B.4 Slave node messages for higher layer servicing (PRM=0, DLS=1)

In the fields of the messages the values are placed using the little endian system. Within an octet or field the bits with increasing number are arranged from right to left.

The first byte of the data block in the messages, defined in this section, contains the following two 4 bit fields:

DP: data priority, which is used during the transmission of the higher layer PDUs to the master node. DP may have one of the following values:
0: lowest priority;
1 to 14: priority 1 to 14;
15: highest priority.

LCN: link channel number. LCN may have one of the following values: 0 to 4 and 8 to 15: reserved for higher layer PDUs;
5: reserved for IPv4-PDUs;
6: reserved for IPv6-PDUs;
7: reserved for DLMS/COSEM-PDUs.

L_FC=0: D_ACK

This message is sent by a slave node as a response to a SEND service expecting a CONFIRM from a higher layer entity.

The structure of the message is shown in Table B.29. The message is sent with CIN ≠ 0xFFFF, LA ≠ 0xFFFF and DBL = 1.

Table B.29 – Structure of the message ‘D_ACK’

Byte of DB	Bit number								Additional description
	7	6	5	4	3	2	1	0	
--	PRM =0	DLS =1	RES =0	DFC	L_FC = 0				LLCF in MPDU header
1	DP				LCN				

L_FC=9: Data response

This message is sent by a slave node as a response to 'Data enquiry' (DLS = 1, L_FC = 11) from the master node. The slave node uses this message to send an non-segmented or segmented higher layer PDU over a specific link channel to the corresponding higher layer entity of the master node. This message supports transmission of up to 16 segments of a segmented higher layer PDU with the maximum total length equal 16 x (DBLMAX – 2) bytes. The segments of the higher layer PDU shall be transmitted in an ascending sequence of messages, starting with the segment number 0.

The message is sent with CIN ≠ 0xFFFF and LA ≠ 0xFFFF. If the slave node does not have data ready for transmission in the requested link channel, then the message is transmitted without a data block (DBL = n = 0).

Table B.30 – Structure of the message 'Data response'

<table>
<tr><th rowspan="2">Byte of DB</th><th colspan="8">Bit number</th><th rowspan="2" colspan="2">Additional description</th></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>--</td><td>PRM =0</td><td>DLS =1</td><td>RES =0</td><td>DFC</td><td colspan="4">L_FC = 9</td><td colspan="2">LLCF in MPDU header</td></tr>
<tr><td>1</td><td colspan="4">DP</td><td colspan="4">LCN</td><td colspan="2"></td></tr>
<tr><td>2</td><td colspan="4">MNS</td><td colspan="4">CSN</td><td colspan="2"></td></tr>
<tr><td rowspan="3">3
to
n [a]</td><td rowspan="3" colspan="8">Higher-layer PDU / segment of higher-layer PDU</td><td>[LSB]</td><td rowspan="3"></td></tr>
<tr><td></td></tr>
<tr><td>[MSB]</td></tr>
<tr><td colspan="11">[a] The maximum value of DBL = n in bytes of frames with DLS = 1 is defined by the parameter DBLMAX.</td></tr>
</table>

The structure of the message is shown in Table B.30, where:

MNS: Maximum number of data segments corresponding to the total number of data segments reduced by one. It has the same value in all messages until the completion of the transmission of all data segments. MNS may have the following values:
- 0: unused value;
- 1 to 15: possible values.

CSN: Current number of the data segment. It may have any values in the range 0 to 15.

Annex C
(informative)

Examples of network scenarios

C.1 General

The following Figure C.1 shows an example of NSC. The p-MN1 communicates as master node with its slave nodes (the s-MN4, s-MN5 and RN6 operating as slave nodes and the slave nodes 7 to 18) and forms a network cell X with them. The p-MN2 and p-MN3 form their own network cells Y and Z.

In each cell at least one communication channel with a fixed channel identification number (CIN) is used. In the whole network at least one common broadcast channel with the CIN=0xFFFF and LA=0xFFFF is used.

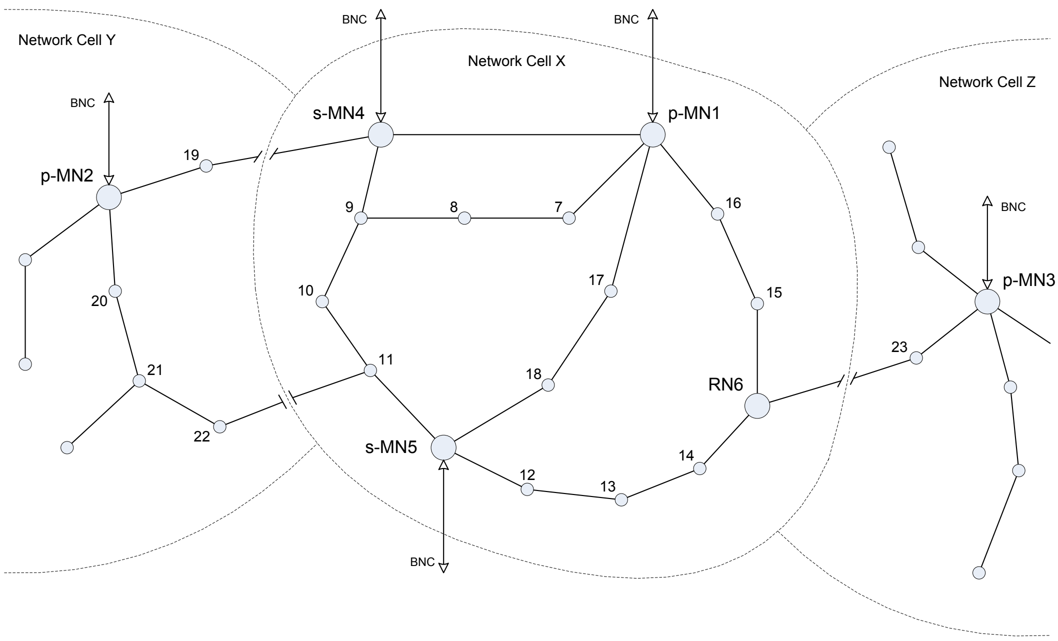

Figure C.1 – Example of NSC

C.2 Examples of an s-MN becoming a master node

An example of an s-MN becoming a master node is given in Figure C.2 below. The p-MN1, p-MN2 and p-MN3 had been operated as master nodes in the cells X, Y and Z. The s-MN5 operated as slave node of p-MN1, with a distance of 3 hops between p-MN1 and s-MN5. Therefore s-MN5 may become a master node, creating its own cell X.2 and increasing the capacity of cell X by splitting it into two cells: X.1 with p-MN1 and X.2 with s-MN5.

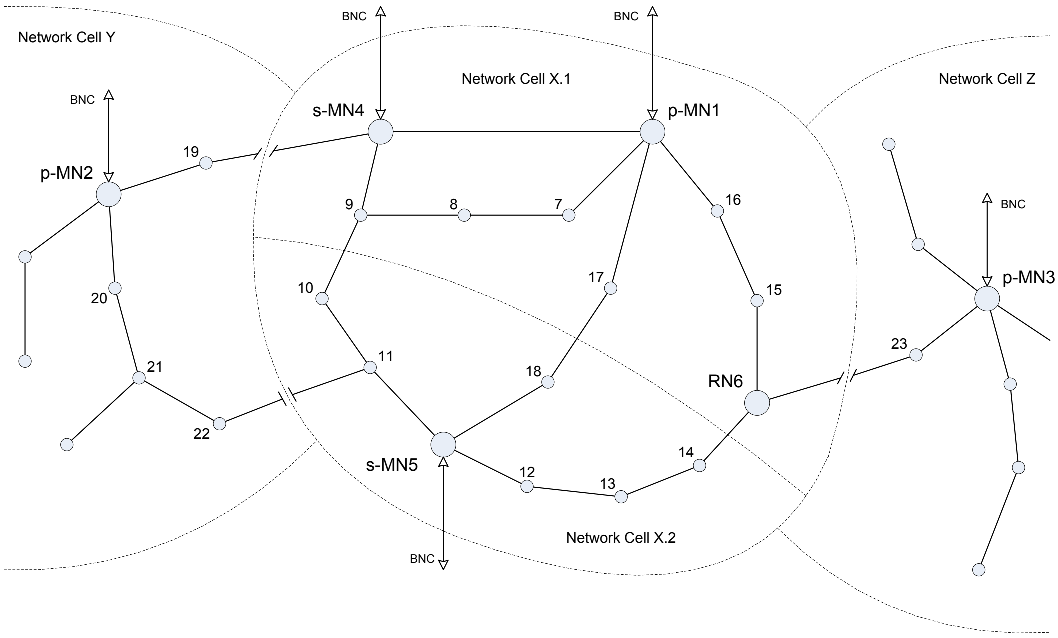

Figure C.2 – Example of an s-MN becoming a master node

If the connection between s-MN4 and p-MN1 is interrupted, s-MN4 will also become a master node after a beacon monitoring period and create its own cell X.3 as shown in Figure C.3 below.

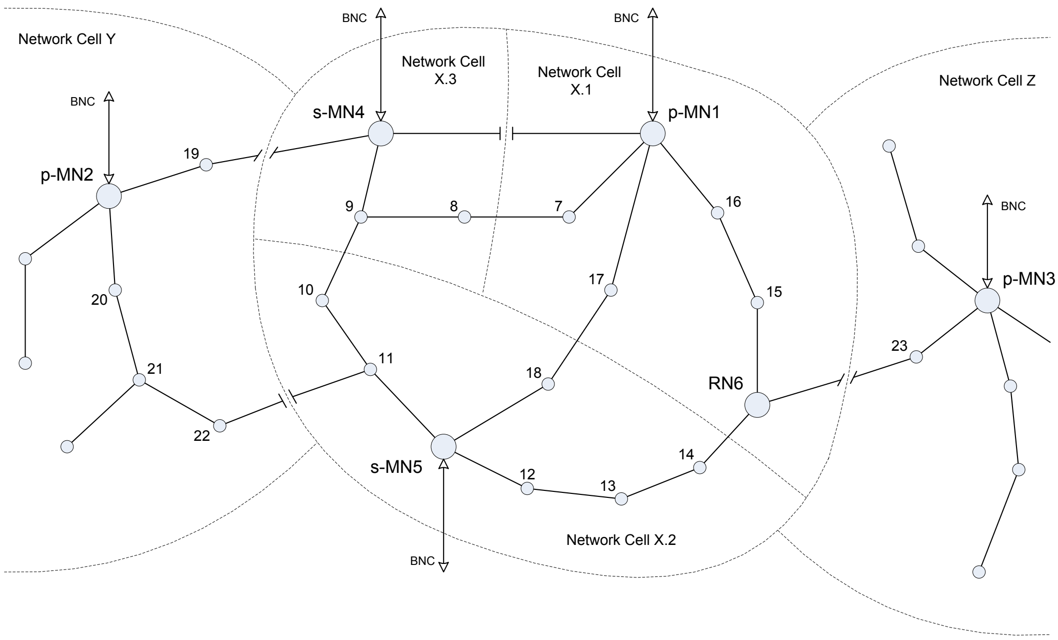

Figure C.3 – Example of cell splitting

Presence of several redundant s-MNs improves the availability of a network. The Figure C.4 below shows an example, where p-MN1 is switched off.

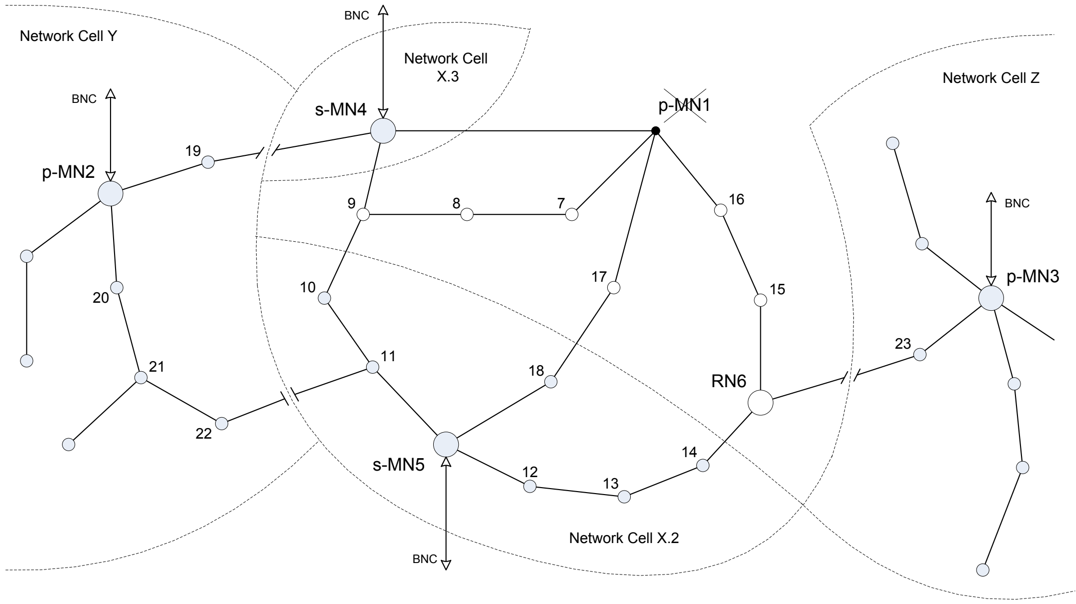

Figure C.4 – Example of switching off a p-MN

Because of missing beacon messages from p-MN1, s-MN4 is becoming a master node creating the cell X.3. Together with the s-MN5, which has been operated as master node within the network cell X.2, it is going to cover the gap. After the two network cells X.3 and X.2 have spread, the gap will disappear (see Figure C.5 below).

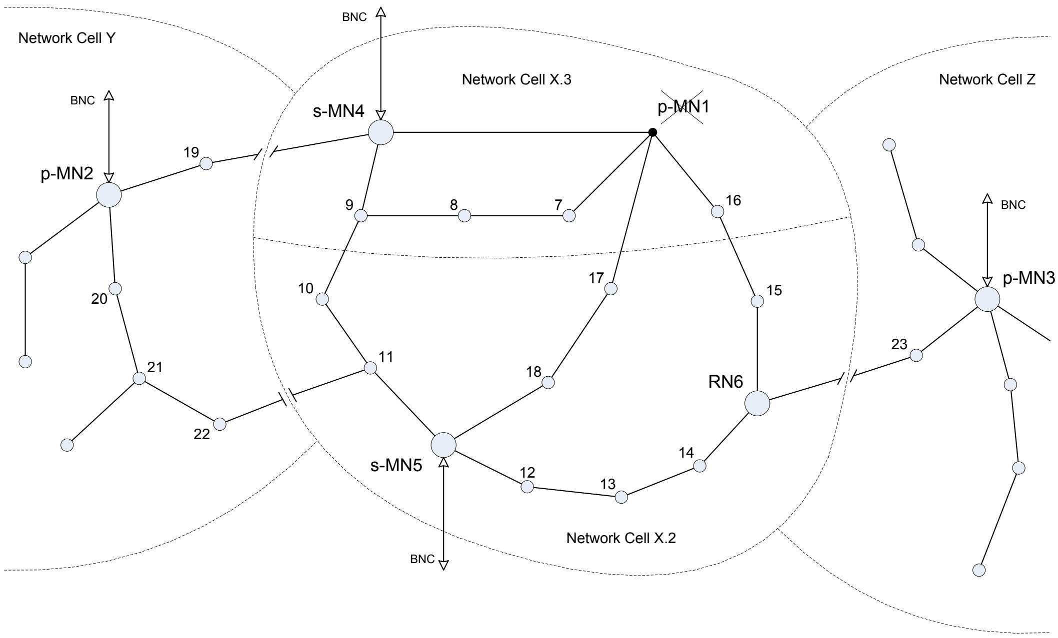

Figure C.5 – Example of cell spreading in NSC

Annex D
(normative)

Configuration and time parameters

The configuration parameter for the PHY layer LLC and MAC sub-layer are shown in Table D.1 and Table D.2 below.

Table D.1 – Configuration parameters for the PHY layer

Parameter name	Data type	Size in bits	Description
ITDD	Integer	16	Initial transmission delay duration in multiples of the sampling interval Δt = 1/347200 s : 0x0000 to 0xFFFF – valid values; 0x0068 – default value.
IFID	Integer	16	Interframe interval duration in multiples of the sampling interval Δt = 1/347200 s : 0x0000 to 0xFFFF – valid values; 0x0558 – default value.
TSL	Integer	8	Length of the training sequence. $(ts\varphi_0,\ ...,ts\varphi_{LTS-1})$ It defines the number of valid PSK encoded elements in the parameter array TSA and shall be an integer multiple of eight. It may have values 8·M, M = 0 to 31. 0 – no training sequence is available; 0x18 – default value.
TSA	Array of 4-bit-integer	248 x 4	Parameter array containing PSK encoded elements of the training sequence $(ts\varphi_0,\ ...,ts\varphi_{LTS-1})$. It defines the phase angel values as an integer multiple of $\pi/8$ radian starting with element $ts\varphi_0$. Each 4-bit-integer may have values 0x0 to 0xF corresponding to the phase angels 0 to $15\pi/8$ radian respectively. The number of valid elements in this array is defined by the parameter TSL. The fist 24 values of the array have the following default values (cf. TSL): 0x8, 0x8, 0x8, 0x8, 0x8, 0x8, 0x8, 0x8, 0x0, 0x8, 0x0, 0x8, 0x0, 0x8, 0x0, 0x8, 0x8, 0x0, 0x8, 0x0, 0x8, 0x0, 0x8, 0x0. The remaining values of array are zero.
CFL0	Array of 24-bit-integer	8 x 24	Carrier frequency list 0 contains values of 8 carrier frequencies f_1 to f_8 in Hz starting with f_1 that are used in the frequency mapping scheme FMS0. Each 24-bit-integer may have values 0x000BB8 to 0xFFFFFF. The array has the following default values in decimal form: 40582, 47345, 54109, 60873, 67636, 74400, 81164, 87927.
CFL1	Array of 24-bit-integer	8 x 24	Carrier frequency list 1 contains values of 8 carrier frequencies f_1 to f_8 in Hz starting with f_1 that are used in the frequency mapping scheme FMS1. Each 24-bit-integer may have values 0x000BB8 to 0xFFFFFF. The array has the following default values in decimal form: 40582, 47345, 54109, 60873, 67636, 74400, 81164, 87927.

CFL2	Array of 24-bit-integer	5 x 24	Carrier frequency list 2 contains values of 5 carrier frequencies f_1 to f_5 in Hz starting with f_1 that are used in the frequency mapping scheme FMS2. Each 24-bit-integer may have values 0x000BB8 to 0xFFFFFF. The array has the following default values in decimal form: 43400, 54250, 65100, 75950, 86800.
CFL3	Array of 24-bit-integer	16 x 24	Carrier frequency list 3 contains values of 16 carrier frequencies f_1 to f_{16} in Hz starting with f_1 that are used in the frequency mapping scheme FMS3. Each 24-bit-integer may have values 0x000BB8 to 0xFFFFFF. The array has the following default values in decimal form: 46113, 48825, 51537, 54250, 56963, 59675, 62387, 65100, 67813, 70525, 73238, 75950, 78663, 81375, 84087, 86800.
CFL4	Array of 24-bit-integer	9 x 24	Carrier frequency list 4 contains values of 9 carrier frequencies f_1 to f_9 in Hz starting with f_1 that are used in the frequency mapping scheme FMS4. Each 24-bit-integer may have values 0x000BB8 to 0xFFFFFF. The array has the following default values: 43400, 48825, 54250, 59675, 65100, 70525, 75950, 81375, 86800
CFL5	Array of 24-bit-integer	8 x 24	Carrier frequency list 5 contains values of 8 carrier frequencies f_1 to f_8 in Hz starting with f_1 that are used in the frequency mapping scheme FMS5. Each 24-bit-integer may have values 0x000BB8 to 0xFFFFFF. The array has the following default values: 48825, 54250, 59675, 65100, 70525, 75950, 81375, 86800.
CFL6	Array of 24-bit-integer	20 x 24	Carrier frequency list 6 contains values of 20 carrier frequencies f_1 to f_7 in Hz starting with f_1 that are used in the frequency mapping scheme FMS6. Each 24-bit-integer may have values 0x000BB8 to 0xFFFFFF. The array has the following default values: 35263, 37975, 40687, 43400, 46113, 48825, 51537, 54250, 56963, 59675, 62387, 65100, 67813, 70525, 73238, 75950, 78663, 81375, 84087, 86800.
CFL7	Array of 24-bit-integer	7 x 24	Carrier frequency list 7 contains values of 7 carrier frequencies f_1 to f_7 in Hz starting with f_1 that are used in the frequency mapping scheme FMS7. Each 24-bit-integer may have values 0x000BB8 to 0xFFFFFF. The array has the following default values: 41457, 49230, 57003, 64776, 72549, 80322, 88096.
CFL8	Array of 24-bit-integer	5 x 24	Carrier frequency list 8 contains values of 5 carrier frequencies f_1 to f_5 in Hz starting with f_1 that are used in the frequency mapping scheme FMS8. Each 24-bit-integer may have values 0x000BB8 to 0xFFFFFF. The array has the following default values: 42514, 53143, 63771, 74400, 85029.

Table D.2 – Configuration parameters for LLC and MAC sub-layers

Parameter name	Data type	Size in bits	Description
MNACTTH	Integer	4	Master node activation threshold in hops: 0x3 to 0xF – valid values; 0x5 – default value. The parameter is only used in a network node, which may operate as master node.
MNDEACTTH	Integer	4	Master node activation threshold in hops: 0x3 to 0xF – valid values; 0x4 – default value. The parameter is only used in a network node, which may operate as master node.
FXTMAX	Integer	4	Maximum value of FXT in a frame: 0x0 to 0x7 – not used; 0x8 – default value; 0x9 to 0xF – other possible values.
DBLMAX	Integer	8	Maximum value of DBL in frames with DLS=1: 0x00 to 0x3F – not used; 0x40 – default value; 0x41 to 0xFF – other possible values.
CRCINITVAL	Integer	16	Initial value for frame header CRC calculation. Valid values: 0x0000 to 0xFFFF. Default value: 0xFFFF.
SCL	Integer	8	Length of scrambling bit sequence (SCL). Valid values: 0x00 to 0xFF. Default value: 0x7F. If SCL=0x00, no scrambling is used.
SCA	Bit array	255	Scrambling bit sequence (SCA) starting with array bit 0. Default value is defined in 6.2.13.
CLCSYNCP	Integer	16	Clock synchronization period in seconds: 0 – no clock synchronization command is sent. 32 to 65535 – other valid values; 900 – default value. The parameter is only used in a network node, which may operate as master node.
M_DLLCTO	Integer	32	Data link layer connection time-out of master node in seconds: 32 to 604800 – valid values; 28800 – default value.
S_DLLCTO	Integer	32	Data link layer connection time-out of slave node in seconds: 32 to 604800 – valid values; 28800 – default value.
S_PDDTH	Integer	32	Power-down duration threshold of slave node in seconds: 32 to 604800 – valid values; 7200 – default value.
M_PDDTH	Integer	32	Power-down duration threshold of master node in seconds: 32 to 604800 – valid values; 2592000 – default value.
DLLCRSTBTH	Integer	16	Data link layer connection re-establishing time-threshold in seconds: 32 to 65535 – valid values; 7200 – default value.
MCTMAX	Integer	16	Maximal duration of a multicast/broadcast transmission in seconds, which may possibly consist of several frames with the same content excepted different TM values: 32 to 65535 – valid values; 70 – default value.

AMCTO	Integer	32	Time-out period for confirmation of a successful acknowledged multicast/broadcast reception in seconds: 32 to 604800 – valid values; 28800 – default value. The parameter is only used in a network node, which may operate as master node.
BRP	Integer	16	Beacon repetition period in seconds: 32 to 65535 – valid values; 900 – default value. The parameter is only used in a network node, which may operate as master node.
BMWT	Integer	16	Beacon monitoring window width in seconds: 32 to 65535 – valid values; 86400 – default value. The parameter is only used in a network node, which may operate as master node.
BTO	Integer	16	Beacon time-out period in seconds: 32 to 65535 – valid values; 7200 – default value. The parameter is only used in a network node, which may operate as slave node.
DLLCNMAX	Integer	16	Maximal number of data link layer connections (i. e. maximal number of slaves registered with the master node): 1 to 2047 – valid values; 2047 – default value; 2048 to 65535 – reserved values. The parameter is only used in a network node, which may operate as master node
S_FXENA	Bit flag	1	Bit flag enabling frame forwarding by slave node: 0 – frame forwarding disabled; 1 – frame forwarding enabled (default value).
S_FXS	Integer	4	Slave Frame Forwarding Sector: 0x0 to 0xE – slave node is assigned to frame forwarding sector 0 to 14 respectively; 0xF – slave node has not been assigned to a dedicated frame forwarding sector (default value). It forwards frames independently of FXS value in a frame.
N_NIN	Integer	16	Network Identification Number of a network node. Valid values: 0x0000 to 0xFFFF. The same value shall be preconfigured in all nodes of a network. 0xFFFF is default value.
M_CIN	Integer	16	Master node channel identification number: 0x0000 to 0xFFFE – regular M_CIN values of a master node; 0xFFFF – no M_CIN value configured (default value). The network node shall not operate as master node. The parameter is only used in a network node, which may operate as master node.
S_CIN	Integer	16	Slave node channel identification number: 0x0000 to 0xFFFE – regular S_CIN values of a slave node; 0xFFFF – no S_CIN value configured (default value). The slave node is not registered with a master node. The parameter is only used in a network node, which may operate as slave node.

S_LA	Integer	16	Link address of slave node: 0x0000 to 0xFFFD – regular S_LA values, that may be assigned to the slave node during its registration with a master node; 0xFFFE – slave node in wait-state waiting for renewal of its registration with the master node, e. g. after a power-down; 0xFFFF – no S_LA value configured (default value). The slave node is not registered with a master node. S_LA may also be changed by the node.
S_NMA	Integer	8	Number of multicast addresses assigned to the slave node (cf. parameter S_MAx). 0x00 – no multicast address has been assigned to the slave node (default value); 0x01 to 0x1F – 1 to 31 multicast addresses have been assigned to the slave node starting with S_MA1; 0x20 to 0xFF – not used. S_NMA may also be changed by the node.
S_MAx	Array of 16 bit integer	31 x 16	Array of 31 sub-parameters starting with S_MA1 to S_MA31, which may contain up to 31 multicast addresses assigned to the slave node. Each of the parameter is 16 bit long and may contain a multicast address value in the range from 0x0000 to 0xFFFD. If the parameter S_NMA has a value n, n = 0 to 30, then each parameter of S_MA_k, to S_MA31, k=n+1 dose not contain an assigned multicast address (i. e.: contains the value 0xFFFF). The value 0xFFFE is not used. Default value of S_MA1 to S_MA31 is 0xFFFF. S_MAx may also be changed by the node.

Bibliography

CLC/TS 52056-8-7:2015, *Electricity metering data exchange – The DLMS/COSEM suite – Part 8-7: AMC-SS PLC communication profile for neighbourhood networks*

EN 60870-5-2, *Telecontrol equipment and systems – Part 5: Transmission protocols – Section 2: Link transmission procedures* (*IEC 60870-5-2)*

This page deliberately left blank

This page deliberately left blank

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Copyright

Useful Contacts:

Customer Services
Tel: +44 845 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 845 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

...making excellence a habit.™