1

- Ce document détaille les specifications des couches basses du CPL G3.
- Il a été soumis auprès du CENELEC (http://www.cenelec.eu) dans le cadre de la normalisation du CPL G3 au titre de “technical specification” (TS). Ce document est issu de la fusion des documents “Spécification de la couche physique CPL G3” et “Spécification de la couche MAC CPL G3”. Des modifications y ont aussi été apportées parmi lesquelles on trouve :
    - Des changements de style imposés par le gabarit des documents sujets à la standardisation au CENELEC,
    - L’ajout de la modulation D8PSK.
- Ce document a servi de base aux travaux du projet européen OPEN meter Project, Topic Energy 2008.7.1.1, Project no.: 226369, www.openmeter.com
- Ce document peut être changé sans préavis.

2

- This document deals with G3-PLC low layers.
- This document has been submitted to CENELEC (http://www.cenelec.eu) for technical specification (TS).This document is the merge of “PLC G3 Physical Layer Specification” and “PLC G3 MAC Layer Specification” with additional changes among which :
    - A change in style due to CENELEC document template,
    - An addition of a modulation scheme : D8PSK.
- This document is based on the results of the European OPEN meter Project, Topic Energy 2008.7.1.1, Project no.: 226369, www.openmeter.com
- This document can be subject to change without prior notice.

3

Version	Date d’application	Titre et nature de la modification	Annule et remplace
1.0		**“Spécification de la couche physique CPL G3”**	
1.0		**“Spécification de la couche MAC CPL G3”**	
2.0	4/04/2011	“**Lower layer profile using OFDM modulation type 2**” • Clarifications • Ajout de la modulation D8PSK • Mise en forme selon le gabarit CENELEC	“Spécification de la couche physique CPL G3” et “Spécification de la couche MAC CPL G3”

4

Electricity metering - Data exchange over powerline

– Part 2: Lower layer profile using OFDM modulation type 2

CONTENTS

List of Figures

List of Tables

INTRODUCTION

This Technical Specification is based on the results of the European OPEN meter Project, Topic Energy 2008.7.1.1, Project no.: 226369, www.openmeter.com.

Electricity metering – Data exchange over powerline – Part 2: Lower layer profile using OFDM modulation Type 2

1 Scope

This standard specifies the Physical and MAC Layer for an Orthogonal Frequency Division Multiplexing (OFDM) Power Line Communications (PLC) system

The physical layer provides a modulation technique that efficiently utilizes the allowed bandwidth within the CENELEC band (3 kHz – 148,5 kHz) thereby allowing the use of advanced channel coding techniques. This combination enables a very robust communication in the presence of narrowband interference, impulsive noise, and frequency selective attenuation.

The medium access control (MAC) layer allows the transmission of MAC frames through the use of the power line physical channel. It provides data services, frame validation control, node association and secure services.

2 Normative references

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

EN 50065-1:2001, *Signalling on low-voltage electrical installations in the frequency range 3 kHz to 148,5 kHz – Part 1: General requirements, frequency bands and electromagnetic disturbances*

IEC 61334-5-1:2001, *Distribution automation using distribution line carrier systems – Part 5-1: Lower layer profiles – The spread frequency shift keying (S-FSK) profile*

IEEE 802:2001, *IEEE Standard for Local and Metropolitan Area Networks – Overview and Architecture*

IEEE 802.15.4:2006, IEE*E Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 15.4: Wireless Medium Access (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*

IEEE 802.2:1998, IEEE *Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 2: Logical Link Control*

IETF RFC 2284: PPP Extensible Authentication Protocol (EAP) [online]. Edited by L. Blunk, J. Vollbrecht. March 1998. Available from: http://www.ietf.org/rfc/rfc2284.txt

IETF RFC 2865: Remote Authentication Dial In User Service (RADIUS) [online]. Edited by C. Rigney, S. Willems, A. Rubens, W. Simpson. June 2000. Available from: http://www.ietf.org/rfc/rfc2865.txt

IETF RFC 3748: Extensible Authentication Protocol (EAP) [online]. Edited by B. Aboda, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz. June 2004. Available from: http://www.ietf.org/rfc/rfc3748.txt

IETF RFC 4291: IP Version 6 Addressing Architecture [online]. Edited by R. Hinden, S. Deering. February 2006. Available from: http://www.ietf.org/rfc/rfc4291.txt

IETF RFC 4764: The EAP-PSK Protocol: A Pre-Shared Key Extensible Authentication Protocol (EAP) Method [online]. Edited by F. Bersani, H. Tschofenig. January 2007. Available from: http://www.ietf.org/rfc/rfc4764.txt

IETF RFC 4862: IPv6 Stateless Address Autoconfiguration [online]. Edited by S. Thomson, T. Narten, T. Jinmey. September 2007. Available from: http://www.ietf.org/rfc/rfc4862.txt

IETF RFC 4944: Transmission of IPv6 Packets over IEEE 802.15.4 Networks [online]. Edited by G. Montenegro, N. Kushalnagar, D. Culler. September 2007. Available from: http://www.ietf.org/rfc/rfc4944.txt

NOTE The following IETF documents are in the draft stage.

IETF draft-daniel-6lowpan-load-adhoc-routing-03: 6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD) [online]. Edited by K. Kim, S. Daniel, G. Montenegro, S. Yoo, N. Kushalnager. June 19, 2007. Available from: http://tools.ietf.org/id/draft-daniel-6lowpan-load-adhoc-routing-03.txt

IETF draft-6lowpan-commissioning-02: Commissioning in 6LoWPAN [online]. Edited by K. Kim, S. Shams, S. Yoo, S. Park, G. Mulligan. July 15, 2008. Available from: http://tools.ietf.org/html/draft-daniel-6lowpan-commissioning-02

IETF draft-thubert-6lowpan-simple-fragment-recovery-02: LoWPAN simple fragment Recovery [online]. Edited by P. Thubert. May 29, 2008. Available from: http://tools.ietf.org/html/draft-thubert-6lowpan-simple-fragment-recovery-02

3 Terms and definitions

For the purposes of this document, the following definitions, together with those of IEEE 802.15.4-2006 and RFC 4944, apply.

Bit fields can be expressed in various notations :

– a field beginning with “0b” is followed by a binary sequence,

– a field beginning with “0h” is followed by a hexadecimal sequence.

4 Acronyms

AAA	Authentication, Authorization, and Accounting
ACK	ACKnowledge
ADP	Adaptation
AFE	Analog Front End
AGC	Automatic Gain Control
BPSK	Binary Phase Shift Keying
CC	Convolutional Code
CENELEC	European Committee for Electrotechnical Standardization
CFA	Contention Free Access
CP	Cyclic Prefix
CRC	Cyclic Redundancy Check
DBPSK	Differential Binary Phase Shift Keying
DQPSK	Differential Quadrature Phase Shift Keying
D8PSK	Differential Eight Phase Shift Keying
FCH	Frame Control Header

FEC	Forward Error Correction
FFT	Fast Fourier Transform
FL	Frame Length
GF	Galois Field
GI	Guard Interval
GMK	Group Master Key
ICI	Inter Carrier Interference
IEEE	Institute of Electrical and Electronics Engineers
IEC	International Electrotechnical Commission
IFFT	Inverse Fast Fourier Transform
IS	Information System
LBD	LoWPAN BootStrapping Device
LBP	LoWPAN Boostrapping Protocol
LSB	Least Significant Bit
LSF	Last Segment Flag
MAC	Media Access Control
MIB	Management Information Base
MPDU	MAC Protocol Data Unit
MSB	Most Significant Bit
NACK	Negative ACKnowledge
NIB	Neighbour Information Base
NSDU	Network Service Data Unit
OFDM	Orthogonal Frequency Division Multiplexing
PAN	Personal Area Network
PAR	Peak to Average Ratio
PDC	Phase Detection Counter
PHY	PHYsical layer
PIB	PAN Information Base
PLC	Power Line Communication
PN	*Pseudo-Noise Sequence*
POS	Personal Operating Space
PPDU	PHY Protocol Data Unit
PPM	Parts Per Million
PSD	Power Spectrum Density
PSDU	PHY Service Data Unit
RADIUS	Remote Authentification Dial in User Service
RC	Repetition Code
RES	Reserved (bit fields)
RMS	Root Mean Square
RS	Reed-Solomon
RX	Receiver
S-FSK	Spread Frequency Shift Keying
S-Robust	Super Robust
SC	Segment Count
SN	Sequence Number

SNR	Signal to Noise Ratio
SYNCP, SYNCM	SYNChronization symbols
TMI	Tone Map Index
TMR	Tone Map Request
TX	Transmit
6LoWPAN	IPv6 over Low power Wireless Personal Area Networks

Furthermore, the abbreviations given in the following clauses apply also:

– Clause 4 of IEEE 802.15.4-2006;

– Clause 1.2 of RFC 4944.

5 Overview

The present standard constitutes the specification for PLC OFDM Type 2 communication based on OFDM modulation, and details the Data Link layer of the protocol stack.

This standard has been developed to meet the following aims:

– Robustness: the communication profile must be suited to severe environments;

– Performance: it must take full advantage of the CENELEC A band;

– Simplicity: it must be simple to implement, install (Plug and Play), operate and maintain;

– Flexibility: it must be compatible with diverse applications and network topologies;

– Security: it must offer a safe environment for the promotion of Value Added services;

– Openness: it must be based on open standards in order to support multi-supplier solutions;

– Scalability: it must support all future metering developments.

To this end, the OFDM PLC protocol stack aggregates several layers and sublayers that form the PLC OFDM Type 2 profile:

– A robust high-performance physical layer based on OFDM and adapted to the PLC environment;

– A MAC sublayer of the IEEE type, well suited to low data rates;

– IPv6, the new generation of IP (Internet Protocol), which widely opens the range of potential applications and services;

– And to allow good IPv6 and MAC interoperability, an adaptation sublayer taken from the Internet world and called 6LoWPAN.

Figure 1 gives an overall view of the PLC OFDM Type 2 communication profile:

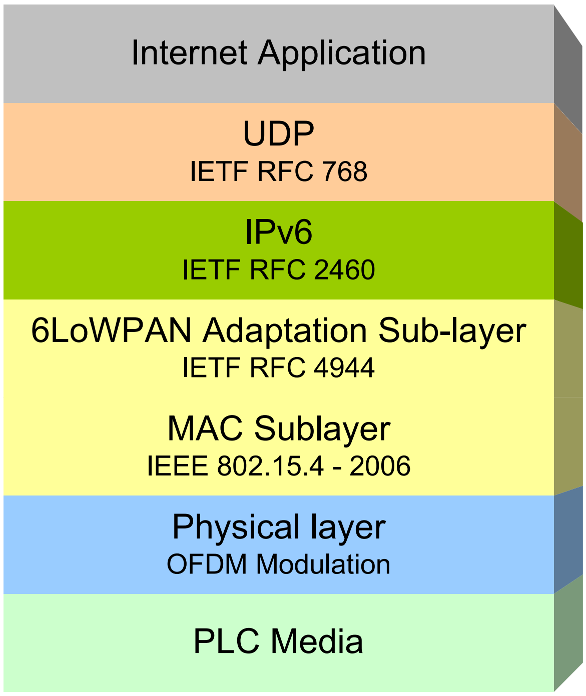


Figure 1 – PLC OFDM Type 2 communication profile

6 Physical layer specification

6.1 Overview of the system

The power line channel is very hostile. Channel characteristics and parameters vary with frequency, location, time and the type of equipment connected to it. The lower frequency regions from 10 kHz to 200 kHz are especially susceptible to interference. Furthermore, the power line is a very frequency selective channel. Besides background noise, it is subject to impulsive noise often occurring at 50/60Hz and group delays up to several hundred microseconds.

OFDM uses advanced modulation and channel coding techniques and thereby it can efficiently utilize the limited bandwidth of CENELEC A band and facilitates a very robust communication over the power line channel.

Figure 2 shows the block diagram of an OFDM system. The allowed bandwidth is divided into a number of sub-channels, which can be viewed as many independent PSK modulated carriers with different non-interfering (orthogonal) carrier frequencies. Convolutional and Reed-Solomon coding provide redundancy bits allowing the receiver to recover lost bits caused by background and impulsive noise. A time-frequency interleaving scheme is used to decrease the correlation of received noise at the input of the decoder, providing diversity.

The OFDM signal is generated by performing IFFT on the complex-valued signal points that are produced by differentially encoded phase modulation (DBPSK, DQPSK and D8PSK) and which are allocated to individual sub-carriers. An OFDM symbol is built by appending a cyclic prefix to the beginning of each block generated by IFFT. The length of cyclic prefix is chosen so that a channel group delay will not cause successive OFDM symbols or adjacent sub-carriers to interfere.

A blind channel estimator technique is used for link adaptation. Based on the quality of the signal received, the receiver decides on the modulation scheme to be used. Moreover, the system differentiates the subcarriers with bad SNR and does not transmit data on them.

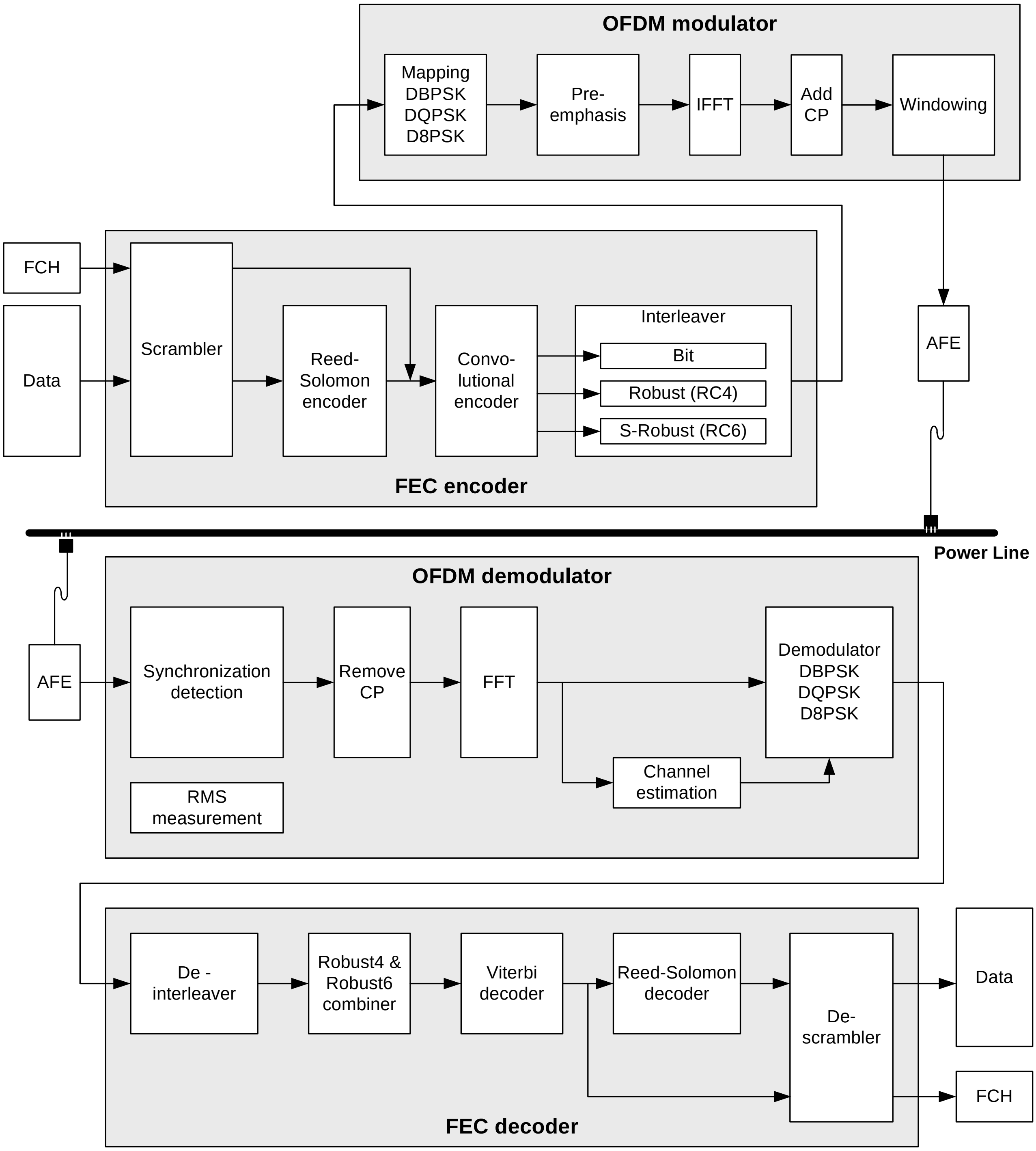


Figure 2 – Block diagram of transceiver

6.2 FEC encoder

6.2.1 Overview

The FEC encoder is composed of a Reed-Solomon encoder followed by a convolutional encoder. In Robust mode, an extra encoder, namely, Repetition Code (RC) is used after the convolutional encoder in order to repeat the bits at the output of convolutional.

6.2.2 Scrambler

The scrambler block helps to give to the data and the FCH a random distribution. The data and FCH stream is ‘XOR-ed’ with a repeating PN sequence using the following generator polynomial:

$$S(x) = x^7 \oplus x^4 \oplus 1$$

This is illustrated in Figure 3:

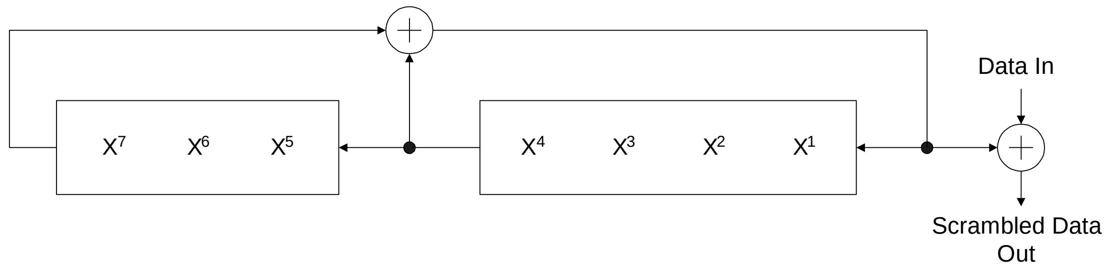


Figure 3 – Data scrambler

The bits in the scrambler are initialised to all ones at the start of processing each physical frame. The scrambler will be reinitialized for FCH and data.

6.2.3 Reed-Solomon encoder

Data from the scrambler is encoded by shortened systematic Reed-Solomon (RS) codes using Galois Field (GF). Depending on the mode used, the following parameter is applied:

– Normal mode - RS (N = 255, K = 239, T = 8) codes using GF (2^8);
– Robust mode - RS (N = 255, K = 247, T = 4) codes using GF (2^8)

Where N, K and T are respectively the total number of symbols transmitted, the number of symbols of data, the number of correctable symbol errors.

The RS symbol word length (i.e., the size of the data words used in the Reed-Solomon block) is fixed at 8 bits. The number of parity words in a RS-block is fixed to 2T bytes.

Code Generator Polynomial g(x) = $\prod_{i=1}^{2T}(x-\alpha^i)$

Field Generator Polynomial p(x) = $x^8 + x^4 + x^3 + x^2 + 1$ (4   35 octal)

The representation of α^0 is “00000001”, where the left-most bit of this RS symbol is the MSB and is first in time from the scrambler and is the first in time out of the RS encoder.

The arithmetic is performed in the Galois Field GF (2^8), where αι is a primitive element that satisfies the primitive binary polynomial $x^8 + x^4 + x^3 + x^2 + 1$.

A data byte (d^7, d^6... d^1, d^0) is identified with the Galois Field element $d^7\alpha^7 + d^6\alpha^6 ... + d^1\alpha + d^0$.

6.2.4 Convolutional encoder

The bit stream at the output of the Reed-Solomon encoder is encoded with a standard rate = ½, K=7 Convolutional encoder. The tap connections are defined as x = 0b1111001 and y = 0b1011011, as shown in Figure 4:

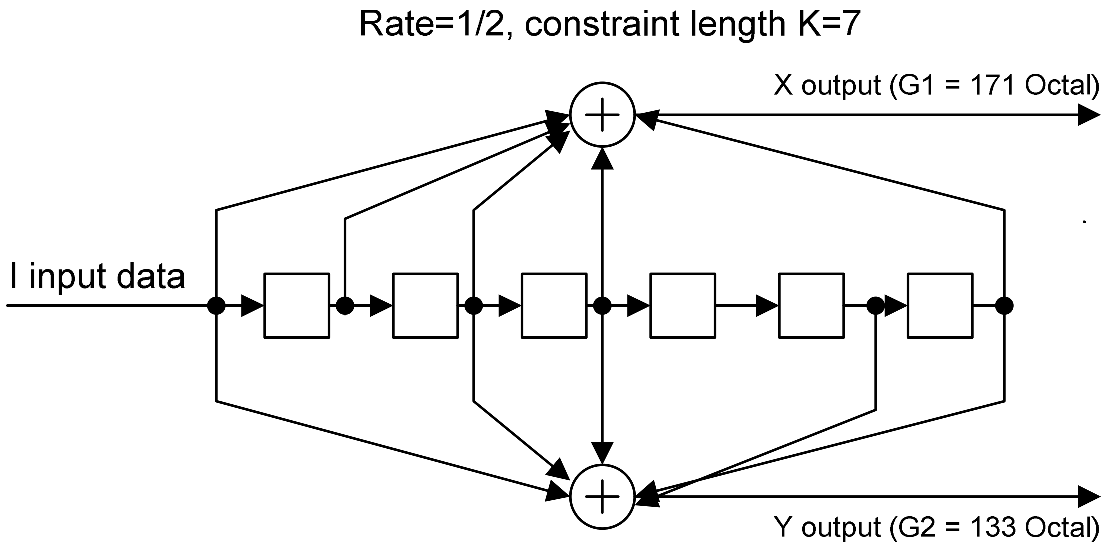


Figure 4 – Convolutional encoder

When the last bit of data to the Convolutional encoder has been received, the Convolutional encoder inserts six tail bits, which are required to return the Convolutional encoder to the "zero state". This reduces the error probability of the Convolutional decoder, which relies on future bits when decoding. The tail bits are defined as six zeros. The bits X and Y are placed in the output stream as XY, X being the first bit leaving the Convolutional encoder.

6.2.5 Robust and Super Robust Modes

6.2.5.1 Robust : Repetition Coding by 4 (RC4)

In Robust mode, the binary stream coming from the convolutional encoder is repeated 4 times before transmission to the interleaver. The repetition is done bit by bit. For an input of '0101', the output will be '0000111100001111'.

NOTE This mode can be used for the data part of the frame only.

6.2.5.2 Super Robust : Repetition Coding by 6 (RC6)

In Super Robust Mode, the binary stream coming from the convolutional encoder is repeated 6 times before transmission to the interleaver. The repetition is done bit by bit. For an input of '0101', the output will be '000000111111000000111111'.

This mode is used for the Frame Control Header (FCH) part of the frame.

6.2.6 Interleaver

The Interleaver is designed such that it can provide protection against two different sources of errors:

- A burst error that corrupts a few consecutive OFDM symbols;
- A frequency deep fade that corrupts a few adjacent frequencies for a large number of OFDM symbols.

To fight both problems at the same time, interleaving is done in two steps. In the first step, each column is circularly shifted a different number of times. Therefore, a corrupted OFDM symbol is spread over different symbols. In the second step, each row is circularly shifted a different number of times, which prevents a deep frequency fade from disrupting the whole column. The amount of circular shifts is determined by the parameters m_i, m_j, n_i, and n_j which are selected based on the number of subcarriers in each OFDM symbol (m) as well as the number of OFDM symbols in each interleaving block (n). Figure 5 shows the order of bits as they are put in the interleaver buffer (i.e. row by row).

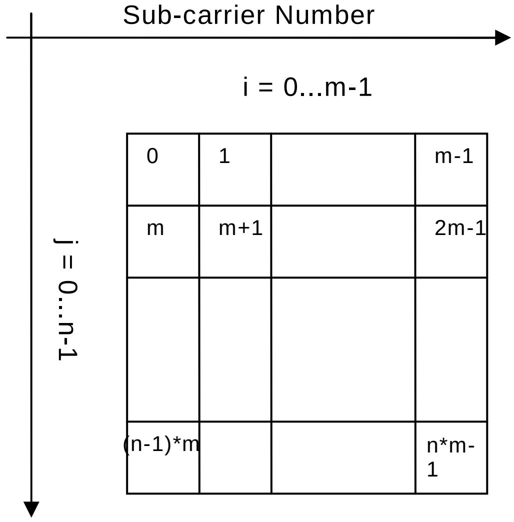


Figure 5 – Interleaver

If the data doesn't completely fill the interleaving matrix, a padding constituted of “zero” bits is added at the end to complete the matrix. The relation between the input and output indexes are determined from the following relations:

Original bit position (i,j) where I = 0, 1,..., m-1 and j = 0, 1,..., n-1

Interleaved position (I,J) where

$J = (j * n_j + i * n_i) \% n$

$I = (i * m_i + J * m_j) \% m$

NOTE % is the modulus operator that gives the remainder of the integer division of one number by another. Ex: 12%5 = 2, 25%3 = 1

NOTE Good interleaving patterns will be generated only if $GCD(m_i,m) = GCD(m_j,m) = GCD(n_i,n) = GCD(n_j,n) = 1$.

A simple search is done to find a good set of parameters based on m and n. Figure 6 displays the spreading behaviour of the Interleaver for n = 8, m = 10, $n_j = 5$, $n_i = 3$, $m_i = 3$ and $m_j = 7$.

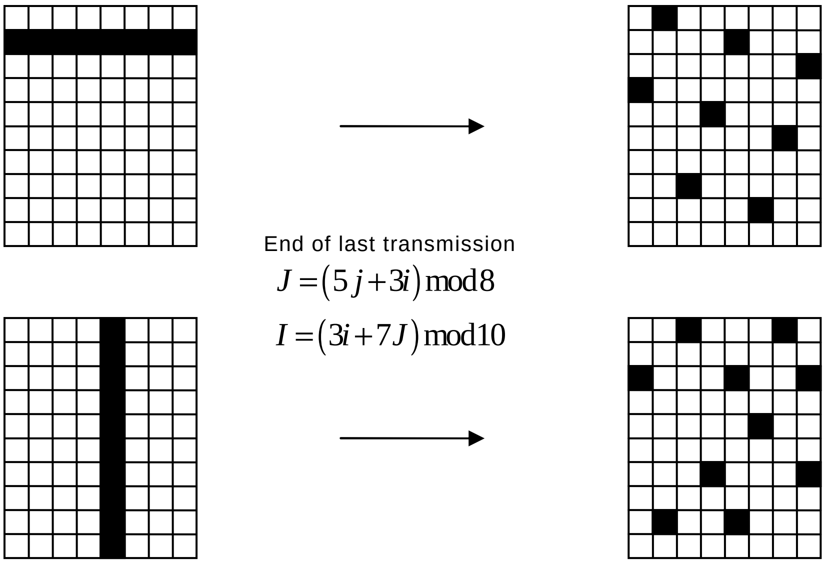


Figure 6 – Spreading behaviour of the Interleaver

To generate a good interleaving pattern, Annex A must be applied.

After interleaving, the mapping functions used for modulation read the output bit in the order defined in Figure 6. If the modulation use M bits per symbol, M bits will be read to compute the modulation for one carrier, so, to form one OFDM Symbol N*m bits are needed.

6.3 OFDM modulator

6.3.1 DBPSK / DQPSK / D8PSK mapping

6.3.1.1 Overview

Each carrier is modulated with Differential Binary or Differential Quadrature or Differential Eight Phase Shift Keying (DBPSK or DQPSK or D8PSK).

The mapping entity is also responsible for assuring that the transmitted signal conforms to the given Tone Map and Tone Mask. The Tone Map and Tone Mask are determined by the MAC layer. The Tone Mask is a predefined (static) system-wide parameter defining the start, stop and notch frequencies. The Tone Map is an adaptive parameter that, based on channel estimation, contains a list of carriers that are to be used for a particular communication between two modems. For example, carriers that suffer deep fades can be avoided, and no information is transmitted on those carriers.

Data bits are mapped for differential modulation (DBPSK, DQPSK, D8PSK or Robust). Instead of using a fixed phase, each phase vector uses the same carrier, previous symbol, as its phase reference. The first FCH symbol uses phase from the last preamble symbol and the first data symbol uses the phase from last FCH symbol. The data encoding for Robust, DBPSK, DQPSK and D8PSK is defined in following clause where Ψ_k is the phase of the k-th carrier from the previous symbol.

6.3.1.2 Mapping for DBPSK and Robust modulations

In DBPSK (and Robust) modulation a phase shift of 0 degrees represents a binary “0” and a phase shift of 180 degrees represent a binary “1” as shown in Table 1.

Table 1 – DBPSK and Robust encoding table of k-th Subcarrier

Input Bit	Output phase

Input Bit	Output phase
0	Ψ_k
1	$\Psi_k + \pi$

Figure 7 shows the constellation diagram of the DBPSK and Robust modulation:

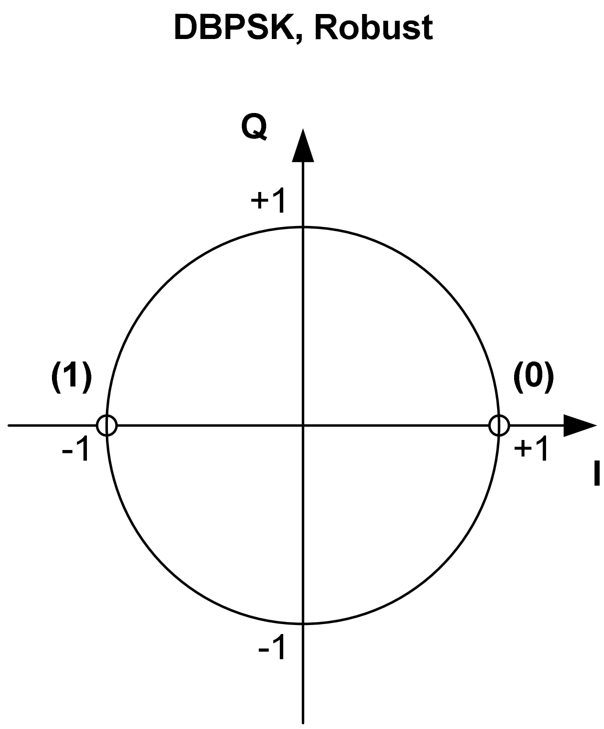


Figure 7 – DBPSK and Robust constellation diagram

6.3.1.3 Mapping for DQPSK modulation

In DQPSK a pair of 2 bits is mapped to 4 different output phases. The phase shifts of 0, 90, 180, and 270 degrees represent binary “00”, “01”, “11”, and “10”, respectively as shown in Table 2.

Table 2 – DQPSK encoding table of k-th subcarrier

Input bit pattern (X,Y), Y is first interleaver output	Output phase
00	Ψ_k
01	$\Psi_k + \pi/2$
11	$\Psi_k + \pi$
10	$\Psi_k + 3\pi/2$

Figure 8 shows the constellation diagram of the DQPSK modulation.

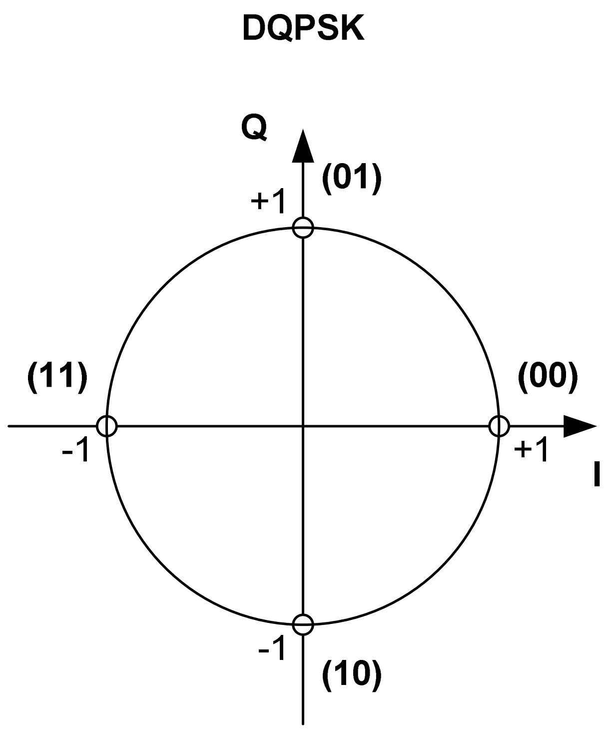


Figure 8 – DQPSK constellation diagram

6.3.1.4 Mapping for D8PSK modulation

In D8PSK a triplet of 3 bits is mapped to one of 8 different output phases. The phase shifts of 0, 45, 90, 135, 180, 225, 270 and 315 degrees represent binary “000”, “001”, “101”, “111”, “011”, “010”, “110” and “100” respectively as shown in Table 3.

Table 3 – DQPSK encoding table of k-th subcarrier

Input bit pattern (X,Y,Z), Z is first interleaver output	Output phase
000	Ψ_k
001	$\Psi_k + \pi/4$
101	$\Psi_k + \pi/2$
111	$\Psi_k + 3\pi/4$
011	$\Psi_k + \pi$
010	$\Psi_k + 5\pi/4$
110	$\Psi_k + 3\pi/2$
100	$\Psi_k + 7\pi/4$

Figure 9 shows the constellation diagram of the D8PSK modulation.

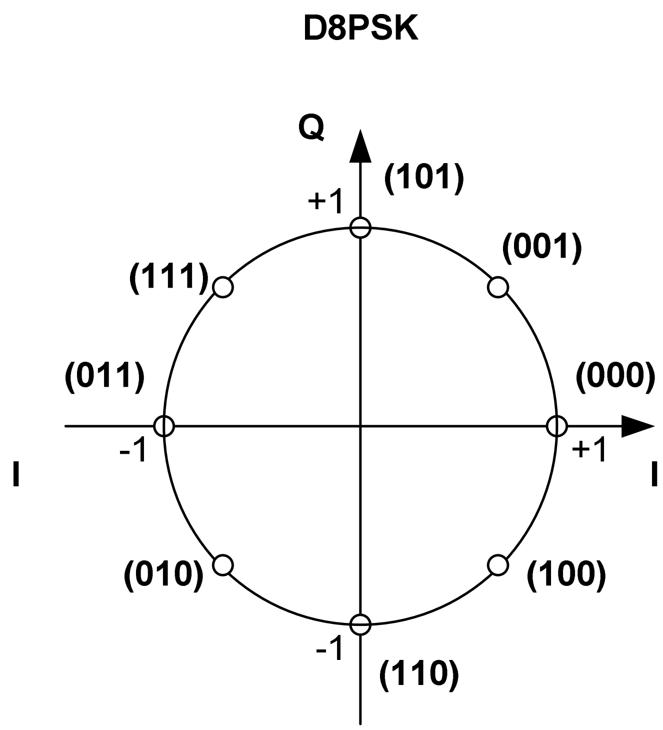


Figure 9 – D8PSK constellation diagram

6.3.2 Frequency domain pre-emphasis

The purpose of this block is to provide frequency shaping to the signal transmitted in order to compensate for the attenuation introduced as the signal travels through the power line.

The frequency-domain pre-emphasis filter shall consist of a multiplier that multiplies the complex frequency domain samples of an OFDM symbol with 128 real filter coefficients. If the optional TXCOEFF parameters are not implemented, the frequency domain pre-emphasis filter should use values to satisfy the spectrum flatness criterion stated in 6.10.5. Otherwise, the filter coefficients are 4 bits representing signed values from -8 to +7. Their values are computed from the TXRES and TXCOEFF parameters that are part of the Tone Map Response message that the destination station sends to the source station. The definition of these values is given in 6.8.

The filter multiplies the first 128 frequency-domain complex samples of an OFDM symbol with the 128 real coefficients of the filter. The rest of the 128 frequency-domain samples of the OFDM symbol shall be set to zero and shall not be multiplied by the filter coefficients. Figure 10 shows the block diagram of the pre-emphasis filter. The output of the filter is the input to the IFFT.

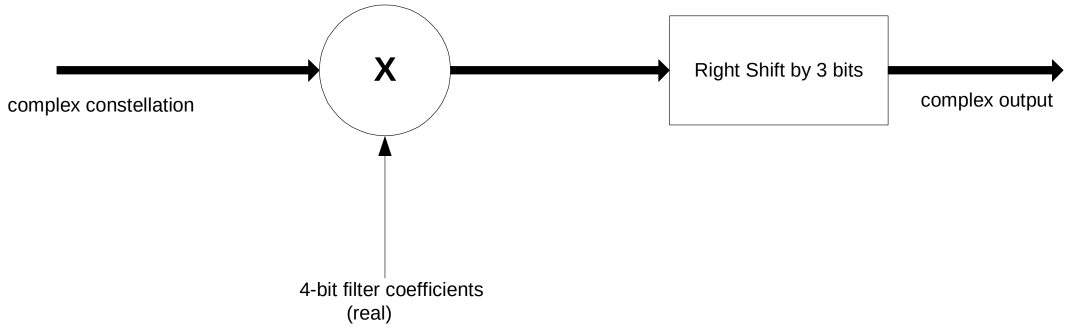


Figure 10 – Block diagram of the pre-emphasis filter

6.3.3 OFDM Generation (IFFT and CP addition)

The OFDM signal is generated using IFFT. The IFFT block takes the 256-point IFFT of the input vector and generates the main 256 time-domain OFDM words pre-pended by 30 samples of cyclic prefix (CP). In other words, we take the last 30 samples at the output of the IFFT and place them in front of the symbol. The useful output is the real part of the IFFT coefficients. The Input/Output configuration is as depicted in Figure 11:

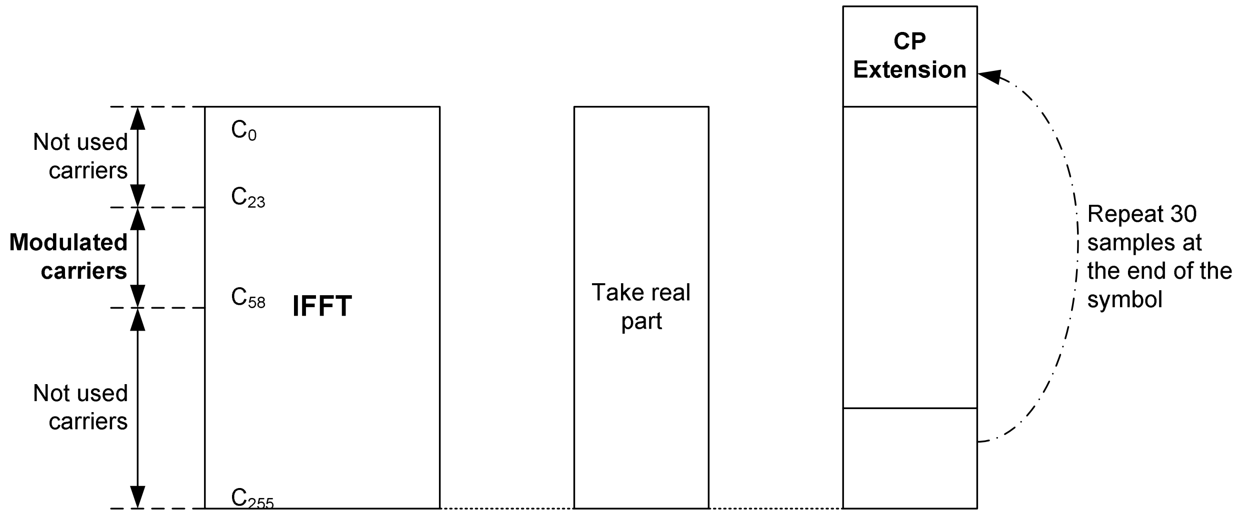


Figure 11 – IFFT Input / Output and CP addition

6.3.4 Windowing

In order to reduce the out of band emission and to reduce the spectral side lobe, the Raised Cosine shaping is applied to all the data symbols. Then the tails and heads of successive symbols are overlapped and added together. This process is described below. Each side of a symbol is first shaped by a Raised Cosine function as shown in Figure 12.

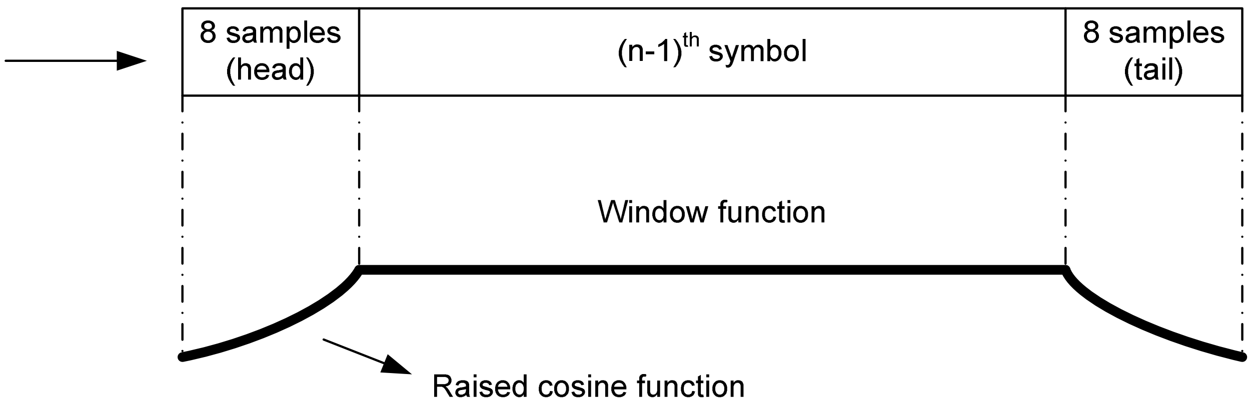


Figure 12 – Raised Cosine windowing

The windowing function at each 8-sample boundary is a Raised Cosine function and its values are given in Table 4. The window function has a value equal to one at all the remaining samples of the symbol. The 8 tail and 8 head shaped samples of the symbol from each side of symbol are overlapped with the tail and head samples of adjacent symbols as shown in Figure 13.

In other words, in order to construct the nth symbol, firstly its 8 head samples are overlapped with the 8 tail samples of the (n 1)th symbol and its 8 tail samples is overlapped with the 8 head samples of the (n+1)th symbol. Finally, the corresponding overlapped parts are added together.

NOTE The head of the first symbol is overlapped with the tail of preamble. The tail of last symbol is sent out with no overlapping applied.

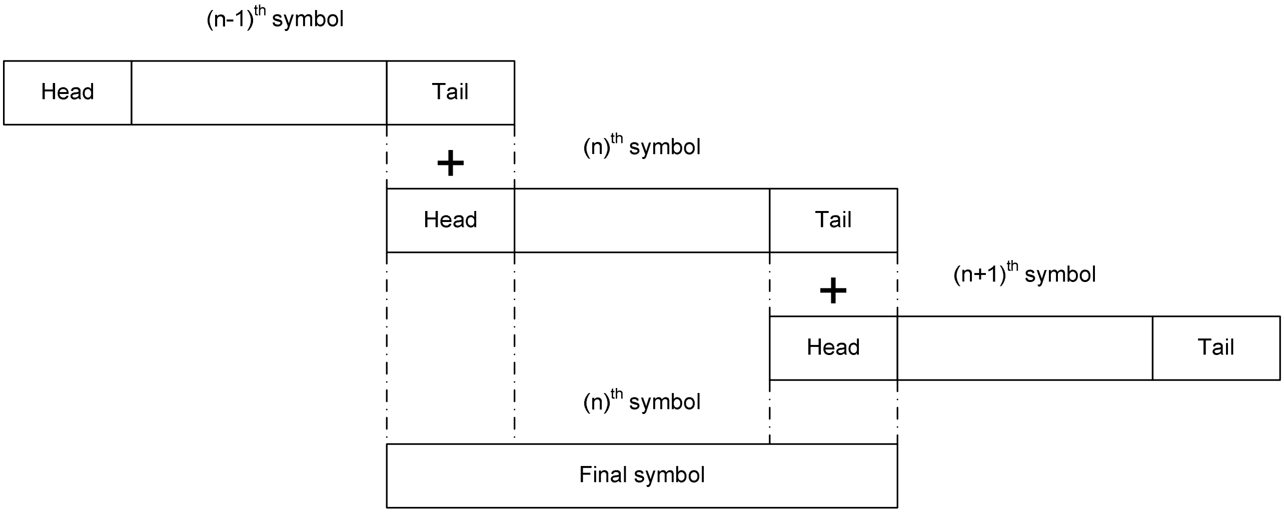


Figure 13 – Overlap / add

Table 4 – The Raised Cosine samples

	Head samples	**Tail samples**
1	0	0,961 9
2	0,038 1	0,853 6
3	0,146 4	0,691 3
4	0,308 7	0,500 0
5	0,500 0	0,308 7
6	0,691 3	0,146 4
7	0,853 6	0,038 1
8	0,961 9	0

6.4 OFDM demodulator

The OFDM demodulator is implementation dependant and out of scope of this standard.

6.5 FEC decoder

The FEC decoder is implementation dependant and out of scope of this standard.

6.6 Structure of physical frames

6.6.1 General

The physical layer supports two types of frames: data and acknowledgment / non-acknowledgment (ACK / NACK) frame.

6.6.2 Physical data frame

A typical data frame for the PLC OFDM Type 2 physical layer is shown in Figure 14:

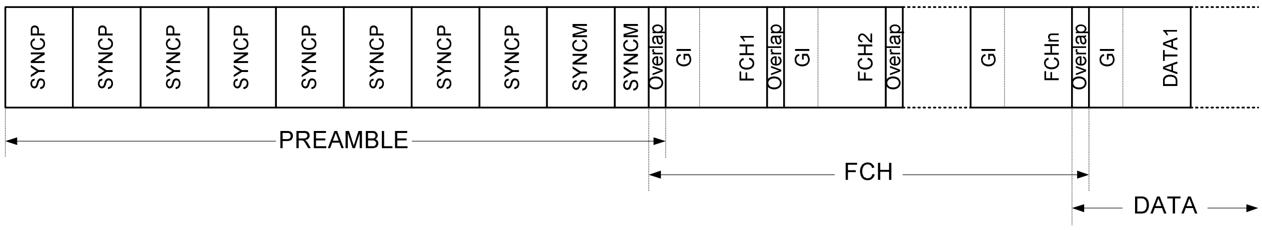


Figure 14 – Typical physical data frame structure

Each frame starts with a preamble, which is used for synchronization and detection in addition to the automatic gain control (AGC) adaptation.

The preamble is followed by n data symbols allocated to Frame Control Header (FCH). The number of symbols depends of the number of carriers used by the OFDM modulation. The different values to be applied in CENELEC A band are defined in 6.7.

FCH carries control information required to demodulate the data frame. Data symbols are transmitted next. In the figure, 'GI' stands for guard interval, which is the interval containing the cyclic prefix.

6.6.3 Physical ACK / NACK frame

The physical layer also supports an ACK/NACK frame, which consists of preamble and the FCH only. The frame structure of the ACK frame is shown in Figure 15 :

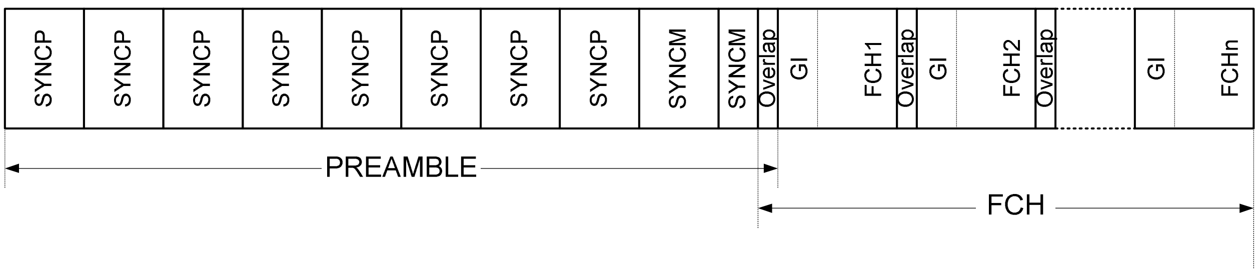


Figure 15 – Example of typical physical ACK / NACK frame structure (content of FCH is submitted to variation of the physical channel)

The bit fields in the FCH explained in 6.6.5 will perform the ACK/NACK signalling.

6.6.4 Preamble

The preamble is composed of 8 identical P symbols and 1½ identical M symbols. SYNCP simply refers to symbols that are multiplied by +1 and SYNCM refers to symbols multiplied by –1. The Preamble consists of eight SYNCP symbols followed by one and a half SYNCM symbols with no cyclic prefix between adjacent symbols. The first symbol includes raised cosine shaping on the leading points. The last half symbol also includes raised cosine shaping on the trailing points. Each of the P and M symbols contain 256 samples and they are pre-stored in the transmitter and transmitted right before the data symbols. The P symbols are used for AGC adaptation, symbol synchronization, channel estimation and initial phase reference estimation. The M symbols are identical to P symbols except that all the carriers are π phase shifted.

At the receiver, the phase distance between symbol P and symbol M waveforms is used for frame synchronization. A P symbol is generated by creating n equally spaced carriers with the phase of each carrier given by the phase vector (ϕ_c). The phase vector definition depends to the number of carriers used. Clause 6.7 describes in detail the different values in case of CENELEC A band.

One way to generate this signal is to start in the frequency domain and create n complex carriers with the initial phase ϕ_c (where n depends to the number of carriers used).

6.6.5 Frame Control Header (FCH)

Immediately after the preamble pattern, the next data symbols are reserved for the frame control header (FCH). Depending of the number of carriers used, the number of symbols varies as defined in next clause.

The FCH is a data structure transmitted at the beginning of each data frame and contains information regarding the current frame. It has information about the type of frame. Table 5 defines the fields of the FCH :

Table 5 – FCH bit fields

Field	Byte	Bit number	Bits	Definition
PDC	0	7-0	8	Phase detection counter see §6.9 AC phase detection
MOD	1	7-6	2	Modulation type: 0 – Robust, 1 – DBPSK, 2 – DQPSK, 3 – D8PSK
FL	1	5-0	6	The frame data length expressed in OFDM symbols (equal to FL*4)
TM[0:7]	2	7-0	8	TM[0:7] – Tone map see §6.7 for binding between tone map bits and carriers
TM[8]	3	7	1	TM[8] – Tone map
DT	3	6-4	3	Delimiter type: 000: Start of frame with no response expected; 001: Start of frame with response expected; 010: Positive acknowledgement (ACK) 011: Negative acknowledgement (NACK) 100-111: Reserved
FCCS	3	3-0	4	Frame Control Check Sequence (CRC5)
	4	7	1	

Field	Byte	Bit number	Bits	Definition
		Total bits	33	

The FCH shall use the default tone map (all allowed subcarriers).

A 5-bit Frame Control Check Sequence (CRC5) is used for error detection in the FCH. The CRC5 is computed as a function of the 28-bit sequence. It is calculated using the following standard generator polynomial of degree 5:

$$G(x) = x^5 + x^2 + 1$$

The CRC5 is the remainder of the division of the FCH polynomial by the generator polynomial. For this calculation, the CRC should be initialized to all ones and the remainder of the division should be inverted (XORed with 11111b). It should be noted that as the CRC5 is not as robust as CRC8, guard bands may be used to make it more robust (such as using reserved values and checking the validity of combination of received values).

NOTE Please note the packing of CRC5 in the FCH as the CRC MSB is packed on the 4th Byte. Please note that the FCH scrambling is performed after the addition of CRC5.

6.7 System fundamental parameters depending CENELEC bands

6.7.1 General specification applied to CENELEC bands

PLC OFDM Type 2 supports the allowed frequencies defined in EN 50065-1.

The DBPSK, DQPSK and D8PSK modulation for each carrier makes the receiver design significantly simpler since no tracking circuitry is required at the receiver for coherently detecting the phase of each carrier. Instead, the phases of carriers in the adjacent symbol are taken as reference for detecting the phases of the carriers in the current symbol.

There is potential to use this standard to support communication in frequencies up to 148,5 kHz. As a result, the sampling frequency at the transmitter and the receiver is selected to be 0,4 MHz in order to provide some margin above the Nyquist frequency for signal filtering in the transmitter (for PSD shaping to remove the signal images) and at the receiver (for band selection and signal enhancement).

The maximum number of carriers that can be used is selected to be 128, resulting in an IFFT size of 256. This results in a frequency spacing between the OFDM carriers equal to 1,5625 kHz * (Fs / N), where Fs is the sampling frequency and N is the IFFT size. Note that imperfections such as sampling clock frequency variation may cause Inter Carrier Interference (ICI). In practice, the ICI caused by a typical sampling frequency variation of about 2 % of the frequency spacing is negligible. In other word, considering ±25 ppm sampling frequency variation in the transmitter and receiver clocks, the drift of the carriers is approximately equal to 8Hz that is approximately 0,5 % of the selected frequency spacing.

The system works in two different modes namely Normal and Robust modes.

In Normal mode, the FEC encoder is composed of a Reed-Solomon encoder (with parity 16 bytes) and a Convolutional encoder.

In Robust mode the FEC encoder is composed of a Reed-Solomon (parity 8 bytes) and a Convolutional encoders followed by a Repetition Code (RC). The RC coder repeats each bit four times making the system more robust to channel impairments. This of course will reduce the throughput by about factor of 4.

The number of symbols in each PHY (Physical Layer) frame is selected based on two parameters, the required data rate and the required robustness.

Table 6 defines the parameter values applied to this standard in the case of CENELEC bands:

Table 6 – Physical parameter values applied to CENELEC bands

Number of FFT points	$N = 256$
Number of overlapped samples	$N_O = 8$
Number of cyclic prefix samples	$N_{CP} = 30$
Sampling frequency	$F_s = 0,4$ MHz
Number of symbols in Preamble	$N_{pre} = 9,5$

The following clauses define specific parameters to be applied for each CENELEC band. As specified in EN 50065-1, the frequency band is divided as described in Table 7:

Table 7 – CENELEC bands

CENELEC band	Frequency range, kHz
A	$3 \le f \le 95$
B	$95 < f \le 125$
C	$125 < f \le 140$
D	$140 < f \le 148,5$

6.7.2 CENELEC A

PLC OFDM Type 2 supports the portion between 35,9 kHz and 90,6 kHz of the CENELEC A band. An OFDM with DBPSK, DQPSK and D8PSK modulation schemes per carrier is selected to support up to 48 kbps data rate in normal mode of operation.

Considering the general parameter values defined in previous clause, the number of usable carriers is given in Table 8.

Table 8 – CENELEC A – Number of carriers

Band	Number of carriers	First carrier (kHz)	Last carrier (kHz)
CENELEC A	36	35,938	90,625

Due to the 36 carriers used for CENELEC A band, the phase vector definition of the preamble and the number of symbols of the FCH are conformant to Table 9 and Table 10:

Table 9 – CENELEC A – Phase vector definition

Carrier	ϕ_c	Carrier	ϕ_c	Carrier	ϕ_c
0	2(π/8)	12	1(π/8)	24	13(π/8)
1	1(π/8)	13	11(π/8)	25	2(π/8)
2	0	14	5(π/8)	26	6(π/8)
3	15(π/8)	15	14(π/8)	27	10(π/8)
4	14(π/8)	16	7(π/8)	28	13(π/8)
5	12(π/8)	17	15(π/8)	29	0
6	10(π/8)	18	7((π/8)	30	2(π/8)
7	7(π/8)	19	15(π/8)	31	3(π/8)

Carrier	ϕ_c	Carrier	ϕ_c	Carrier	ϕ_c
8	3(π/8)	20	6(π/8)	32	5(π/8)
9	15(π/8)	21	13(π/8)	33	6(π/8)
10	11(π/8)	22	2(π/8)	34	7(π/8)
11	6(π/8)	23	8(π/8)	35	7(π/8)

Table 10 – CENELEC A - Number of FCH symbols

	Number of FCH symbols, N_{FCH}
CENELEC A	13

Given the FCH data length is 33 bits, there are 13 FCH symbols required. This can be calculated using:

Number of FCH Symbols = ceiling (((N_FCH_bits + 6) x 2 x 6)/36) = ceiling ((39 x 2 x 6)/36) = 13.

NOTE Ceiling(*x*) = is the smallest integer not less than *x*.

The number of symbols, Reed-Solomon block sizes, and data rate associated with 36 carriers is tabulated in Table 11 and Table 12. Table 13 shows the data rate including the data transmitted in FCH. To calculate the data rate, it is assumed that the packets are continuously transmitted with no inter-frame time gap.

Table 11 – CENELEC A – RS block size for various modulations

CENELEC A	RS block size (Out/In) per modulation type, bytes			
Number of symbols	D8PSK, P16 [1)]	DQPSK, P16 [1)]	DBPSK, P16 [1)]	Robust, P8 [2)]
12	(80/64)	(53/37)	(26/10)	N/A
20	(134/118)	(89/73)	(44/28)	N/A
32	(215/199)	(143/127)	(71/55)	N/A
40	N/A	(179/163)	(89/73)	(21/13)
52	N/A	(233/217)	(116/100)	(28/20)
56	N/A	(251/235)	(125/109)	(30/22)
112	N/A	N/A	(251/235)	(62/54)
252	N/A	N/A	N/A	(141/133)

[1)] P16 is Reed-Solomon with 16 bit parity

[2)] P8 is Reed-Solomon with 8 bit parity

NOTE N/A means not applicable and the reason is that the corresponding number of symbols specified results in RS encoder block length that exceeds the maximum allowable limit of 255.

Table 12 – CENELEC A – Data rate for various modulations (excluding FCH)

CENELEC A	Data rate per modulation type, bps			
Number of Symbols	D8PSK, P16 [1)]	DQPSK, P16 [1)]	DBPSK, P16 [1)]	Robust, P8 [2)]
12	21 829	12 103	3 271	N/
20	32 534	19 456	7 462	N/A
32	42 618	26 489	11 471	N/A
40	N/A	29 693	13 298	2 423
52	N/A	33 221	15 309	3 121

CENELEC A	Data rate per modulation type, bps			
Number of Symbols	D8PSK, P16 [1]	DQPSK, P16 [1]	DBPSK, P16 [1]	Robust, P8 [2]
56	N/A	34 160	15 844	3 257
112	N/A	N/A	20 009	4 647
252	N/A	N/A	N/A	5 592

[1] P16 is Reed-Solomon with 16 bit parity

[2] P8 is Reed-Solomon with 8 bit parity

NOTE N/A means not applicable and the reason is that the corresponding number of symbols specified results in RS encoder block length that exceeds the maximum allowable limit of 255.

Table 13 – CENELEC A – Data rate for various modulations (including FCH)

CENELEC A	Data rate per modulation type, bps			
Number of Symbols	D8PSK, P16 [1]	DQPSK, P16 [1]	DBPSK, P16 [1]	Robust, P8 [2]
12	23 235	13 453	4 620	N/A
20	33 672	20 556	8 562	N/A
32	43 501	27 349	12 332	N/A
40	N/A	30 445	14 049	3 192
52	N/A	33 853	15 941	3 765
56	N/A	34 759	16 444	3 867
112	N/A	N/A	20 360	5 002
252	N/A	N/A	N/A	5 765

[1] P16 is Reed-Solomon with 16 bit parity

[2] P8 is Reed-Solomon with 8 bit parity

NOTE N/A means not applicable and the reason is that the corresponding number of symbols specified results in RS encoder block length that exceeds the maximum allowable limit of 255.

The data rate is calculated based on the number of symbols per PHY frame (N_S), the number of carriers per symbol (N_{carr}) and the number of parity bits added by FEC blocks.

As an example, consider the system in the CENELEC A band working in Robust mode with 40 symbols of data. The total number of bits carried by the whole physical frame is equal to:

$$Total\_No\_Bits = N_S \times N_{carr} = 40 \times 36 = 1\,440 \text{ bits}$$

The number of bits required at the input of the Robust encoder is given by:

$$No\_Bits\_Robust = 1\,440 \times Robust_{Rate} = 1\,440 \times ¼ = 360 \text{ bits}$$

NOTE Due to the fact that the Robust mode reduces the throughput by about factor 4, the rate is also reduced by about factor 4.

Considering the fact that the convolutional encoder has a rate equal to ½ (CCRate = ½) and also consider adding CCZerotail = 6 bits of zeros to terminate the states of the encoder to all zero states then the maximum number of symbols at the output of the Reed-Solomon encoder ($MAXRS_{bytes}$) is equal to:

$$MAXRSbytes = floor((No\_Bits\_Robust \times CC_{Rate} - CCZeroTail)/8) = floor((360 \times ½ - 6)/8) = 21$$

NOTE floor(x) is the largest integer not greater than x

According for eight symbols associated with the parity bits (in Robust mode), we obtain:

$$DataLength = (21 - ParityLength) \times 8 = 104 \text{ bits}$$

These 104 bits are carried within the duration of a physical frame. The duration of a physical frame is calculated by the following formula:

$$T_{Frame} = ((NS+N_{FCH}) \times (N_{CP} + N - N_{O}) + (N_{pre} \times N))/F_s$$

Where:

- NS is the number of symbols to transmit;
- N_{pre} is the number of symbols in the preamble;
- N is the FFT length;
- N_O is the number of samples overlapped at each side of one symbol;
- N_{CP} is the number of samples in the cyclic prefix;
- N_{FCH} is the number of symbols in the FCH;
- F_s is the sampling frequency (in Hz).

With the actual values of the parameters the physical frame duration is equal to:

$$T_{Frame} = ((40+13) \times (30 + 256 - 8) + (9.5 \times 256))/ 400\,000 = 0{,}043 \text{ s.}$$

Therefore the data rate is:

$$\text{Data rate} = 104 / 0.042 \sim 2{,}4 \text{ kbps.}$$

6.8 Adaptive tone mapping & transmit power control

6.8.1 General

PLC OFDM Type 2 shall estimate the SNR of the received signal sub-carriers (tones) and adaptively select the usable tones, the optimum modulation and coding type (including DBPSK, DQPSK, D8PSK and the Robust mode) to ensure reliable communication over the power line channel. It shall also specify what power level the remote transmitter shall use and what gain values it should apply for the various sections of the spectrum. The per-carrier quality measurement enables the system to adaptively avoid transmitting data on sub-carriers with poor quality. Using a tone map indexing system, where the index is passed from receiver to transmitter and vice versa, allows the receiver to adaptively select which sub-carriers will be used for transmission and which ones will be used to send dummy data that the receiver will ignore.

The goal of the adaptive tone mapping is to allow the PLC OFDM Type 2 receiver to achieve the greatest possible throughput under the given channel conditions between the transmitter and the receiver. In order to accomplish this goal, the receiver shall inform the remote transmitter which tones it should use to send data bits on, and which tones it should use to send dummy data bits that the receiver shall ignore. The receiver shall also inform the remote transmitter how much amplification or attenuation it should apply to each of the tones.

The transmitter may request the receiver to estimate a channel condition by setting the TMR bit of the Segment Control header of the MAC layer, as described in 7.3.3.2.2.

The receiver has to estimate this particular communication link between the two points and choose optimal PHY parameters. This information will be sent back to the originator as a

Tone Map Response. Note that for broadcast always the Robust mode will be used. The other modulation schemes are used for point-to-point or multi-point.

6.8.2 PN Modulating un-used subcarriers

The mapping function for DBPSK, DQPSK, D8PSK and Robust must obey the Tone Mask, thus carriers that are masked are not assigned phase symbols, and the amplitude is zero. When the modulation type is DBPSK, DQPSK or D8PSK the mapping function also obeys the Tone Map. When a carrier is encountered on which no information is to be transmitted, the mapping function substitutes a binary value from a Pseudo Noise (PN) sequence. The binary value shall be used as the value for both bits in the case of DQPSK and for 3 bits for D8PSK.

The PN sequence shall be generated using the same generator polynomial introduced in 6.2.2. The bits in the PN sequence generator shall all be initialized to ones at the start of processing each frame and sequenced to the next value after every mapped, unmapped or masked carrier. The first value of the PN sequence (the output when all bits are initialized to ones) corresponds to carrier number “0” of the first OFDM symbol of each frame and the 37th value corresponds to carrier number “0” of the second OFDM symbol.

6.9 AC phase detection

This is a basic approach to detect AC phase associated to a device, however the optimal design would be dependent on the particular network topology. This information is mainly useful at system level in order to check for unexpected losses on the distribution line and must be stored locally.

Three phases on the mains are sinusoidal waveforms with a phase shift of 120° from each other where each half cycle is equal to 10 ms at 50 Hz and 8,3 ms at 60 Hz. A zero-crossing detector delivers an output pulse based on the signal transition through zero volt of a 50/60 Hz sinusoidal on the power line, and is used to synchronize a Tx-device and an Rx-device. The Tx-device generates a time stamp based on an internal counter at the instant a packet shall be transmitted. The receiver provides its own time stamp, and the difference between the Tx-device and the Rx-device time stamps provides the phase difference. The procedure to achieve the phase difference between transmitter and receiver is as follows.

a) All devices shall have internal timers, which are synchronized to zero-crossing detector.

b) All devices shall have a zero-crossing detector that delivers an output pulse such that the pulse width is 5 % of the total period. The characteristic of the zero-crossing detector is shown in Figure 16:

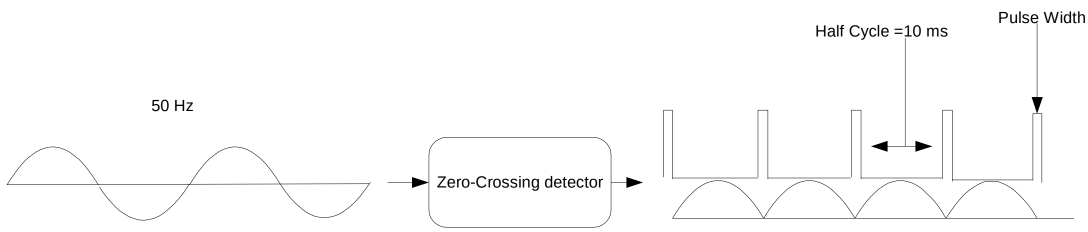


Figure 16 – Zero-crossing detector

c) An eight bits counter provides a time stamp placed on the FCH frame upon transmission of payload.

d) Upon detection of the FCH frame, the receiver shall compute the delay, which is the difference between transmit counter (PDC) and receive counter. The phase differential shall be computed shown below.

e) Phase differential = (Rx_counter - Tx_Counter) /3

Electromagnetic propagation time and additional delay for packet processing and detection shall be considered measuring delay. The electromagnetic propagation delay is 5,775 µs/km, which is negligible. However, a processing delay, comprising the transmission_delay and detection_delay shall be factored into the equation above as follows.

New_Phase differential = ((Rx_counter – detection_delay) – (Tx_Counter – trasmission_delay))/3.

6.10 Transmitter electrical specifications

6.10.1 Output level measurement

PLC OFDM Type 2 transmitter output level shall be compliant with EN 50065-1. No part of the spectrum of the signal shall exceed 120 dB µV.

The output level is measured with a peak detector (with 200 Hz bandwidth).

6.10.2 Transmit spectrum mask (frequency notching)

6.10.2.1 General

PLC OFDM Type 2 PHY is provisioned to have programmable notches at certain frequencies in order to:

- Avoid certain frequencies that are reserved by power line regulatory bodies for other applications;
- Allow cohabitation with PLC S-FSK systems in compliance with IEC 61334-5-1;
- Allow cohabitation with other potential systems operating on power line.

Depending on the relative position of the required notch frequency to ensure cohabitation, a few sub-carriers are masked. No data is sent over the masked sub-carriers. According to Figure 17, if the notch frequency is in the R1 region, SC(n-1), SC(n) and SC(n+1) are masked (total three sub-carriers). If the notch frequency is in the R2 region the two nearest sub-carriers in either side (i.e. SC(n-1), SC(n), SC(n+1) and SC(n+2)) are masked (a total of four sub-carriers).

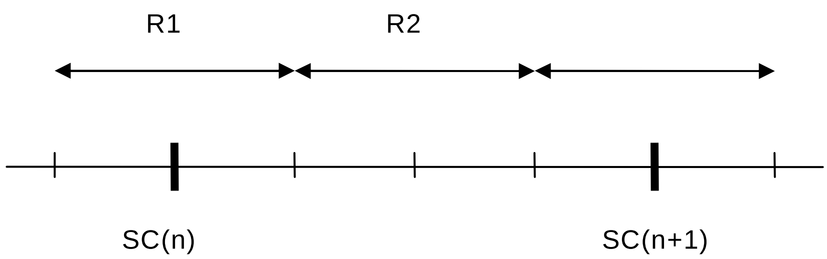


Figure 17 – Transmit spectrum mask

The notching map should be a global parameter that is set in the initialization step of the devices. As described above, to provide sufficiently deep notches for a particular frequency band, it is required to zero one (or sometimes two) extra sub-carriers before and after that band, depending on the position of the notch with respect to the sub-carriers. The following pseudo code can be used for the decision between one/two extra sub-carriers.

```
if NotchFreq / SamplingFreq * FFTSize is in R1

        Sc(n-1) = Sc(n) = Sc(n+1) = 0;
```

```
if NotchFreq / SamplingFreq * FFTSize is in R2
        Sc(n-1) = Sc(n) = Sc(n+1) = Sc(n+2) = 0;
```

Where, Sc is an array that determines which sub-carriers are used to transmit data (if Sc(i) is zero, no data is sent using that sub-carrier).

Frequency notching reduces the number of active tones that are used for transmitting information. Since notching is done for all the transmit signals, including the Frame Control (FC) data, the number of symbols in FC depends on the number of active carriers.

6.10.2.2 Cohabitation with S-FSK systems

In order to cohabitate with S-FSK systems, the transmitter shall use an appropriate scheme to insert deep notches in the spectrum. In particular, the two frequencies referred to in the IEC 61334-5-1 standard as mark and space frequencies f_M and f_S, shall be notched.

In order to have minimum effect on S-FSK, the OFDM modem working in CENELEC A band, shall not transmit any signal between S-FSK frequencies i.e. in 63 kHz to 74 kHz band. The notched subcarriers in this mode are shown in Table 14:

Table 14 – Notched subcarriers in S-FSK cohabitation mode

Sub-carrier number	Frequency of the sub-carrier
39	60,937 5
40	62,500 0
41	64,062 5
42	65,625 0
43	67,187 5
44	68,750 0
45	70,312 5
46	71,875 0
47	73,437 5
48	75,000 0
49	76,562 5

With this, 11 sub-carriers cannot transmit data. Considering the fact that there are a total of 36 carriers available, 25 sub-carriers remain for data transmission, resulting in FC with 19 OFDM symbols because:

$$\text{Number of OFDM symbols} = \text{ceiling}\left((33 + 6) * 2 * 6 / 25\right) = 19.$$

NOTE Ceiling(*x*) = is the smallest integer not less than *x*.

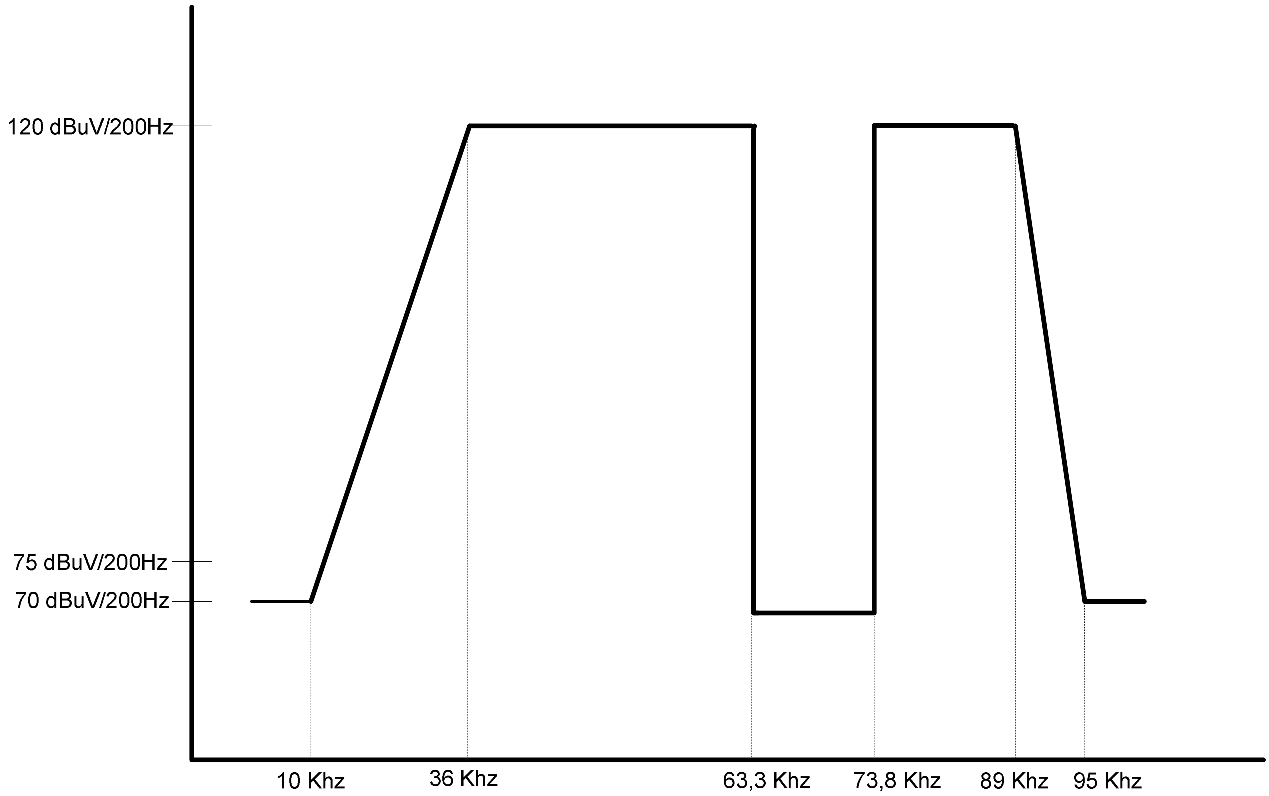


Figure 18 – Spectrum with two notches inserted to cohabitate with S-FSK PLC modem

All devices shall use tone masking on the carriers specified in each S-FSK device in order to be compliant with the transmit spectrum mask shown in Figure 18. The transmitted power spectral density of notched frequency shall be 25 dB below the limits specified for the remaining sub-carriers.

In order to verify compliance to the above requirement, measurements should be made using a spectrum analyzer with a resolution bandwidth set at 200 Hz and a quasi-peak detector. The transmitter shall be configured to repeatedly transmit frames with lengths equal to the maximum length allowed in normal mode.

6.10.3 Spurious transmission

It is the obligation of the manufacturer to ensure that spurious transmissions conform to regulations in effect for the country in which this station is used.

6.10.3.1 System clock frequency tolerance

The system clock tolerance shall be ± 25 ppm maximum. The transmit frequency and symbol timing shall be derived from the same system clock oscillator.

6.10.4 Transmit constellation accuracy

6.10.4.1 Transmit constellation error

The relative constellation RMS error, averaged over all subcarriers in a symbol, and averaged over several OFDM symbols, shall not exceed -15 dB from the ideal signal RMS level.

6.10.4.2 Transmit modulation accuracy test

The transmit modulation accuracy test shall be performed by instrumentation capable of converting the transmitted signal into a stream of samples, using at least the sampling rate applicable for the CENELEC A band, with sufficient accuracy in terms of amplitude, DC offsets, and phase noise. The sampled signal shall be processed in a manner similar to an actual receiver, according to the following steps, or an equivalent procedure.

An example is provided below:

a) Pass a sequence of 88 bytes all ones, representing a 12-symbol QPSK frame, through an ideal floating point *pseudo-transmitter,* for example using matlab, and save the complex IFFT input for each of the 12 data symbols as $A_{ic}\exp[j\Phi_{ic}]$, where i is the symbol number and c is the carrier number corresponding to that symbol. 'i' will have values between 0 and 11 while c will be between 0 and 35. The ideal sub-transmitter should include all the transmitter blocks specified in this standard, including Scrambler, RS encoder, convolutional encoder, interleaver and mapper.

b) Next, use the transmitter under test to generate the same frame using the bits specified in step a).

c) Connect the test equipment that will simulate the receiver directly to the transmitter to detect start of frame.

d) Save all 12 data symbols of the frame.

e) Offline, apply a floating-point FFT on each symbol and store the complex values as $B_{ic}\exp[j\theta_{ic}]$ where i is the symbol number and c is the carrier number corresponding to that symbol.

f) Compute the RMS error between the transmitted and ideal constellation points for each symbol as the sum of the squared Euclidean distance between the two points over all the carriers in the symbol:

$$error\_rms_i = \sum_{c=0}^{35} abs\{Aic * \exp[j\Phi ic] - Bic * \exp[j\Theta ic]\}^2$$

Next compute the total RMS error as the sum of the RMS errors of the individual symbols:

$$total\ error\ rms = \sum_{i=0}^{11} error\_rms_i$$

Compute the RMS of each transmitted symbol as:

$$Tx\_rms_i = \sum_{c=0}^{35} Aic^2$$

And the total RMS for all transmitted symbols as:

$$total\ Tx\ rms = \sum_{i=0}^{11} Tx\_rms_i$$

g) Total error RMS should satisfy the following equation:

20*log10 (total error rms / total Tx rms) < -15 dB

6.10.5 Transmitter Spectral Flatness

No individual carrier shall have average power outside of the range +/- 2 dB with respect to the average power in all of the carriers.

The measurement shall be performed into 50Ω impedance.

6.11 Physical Layer Primitives

6.11.1 Data primitives

This clause describes the different data primitives accessible between the MAC and PHY layers.

The transmission protocol between the MAC and the PHY layer shall be done as follow (see also Figure 19):

a) The receipt of the PD-DATA.request primitive by the PHY entity will cause the transmission of the supplied PSDU to be attempted;
b) The PHY will first construct a PPDU, containing the supplied PSDU, and then transmit the PPDU;
c) When the PHY entity has completed the transmission, it will issue the PD-DATA.confirm primitive with a status of SUCCESS.

NOTE 1 At PHY level, if a PD-DATA.request primitive is received while the receiver is enabled (RX_ON state), the PHY entity will discard the PSDU and issue a PD-DATA.confirm primitive with a status of FAILED.

NOTE 2 At PHY level, if a PD-DATA.request primitive is received while the transmitter is already busy transmitting (BUSY_TX state), the PHY entity will discard the PSDU and issue a PD-DATA.confirm primitive with a status of BUSY_TX.

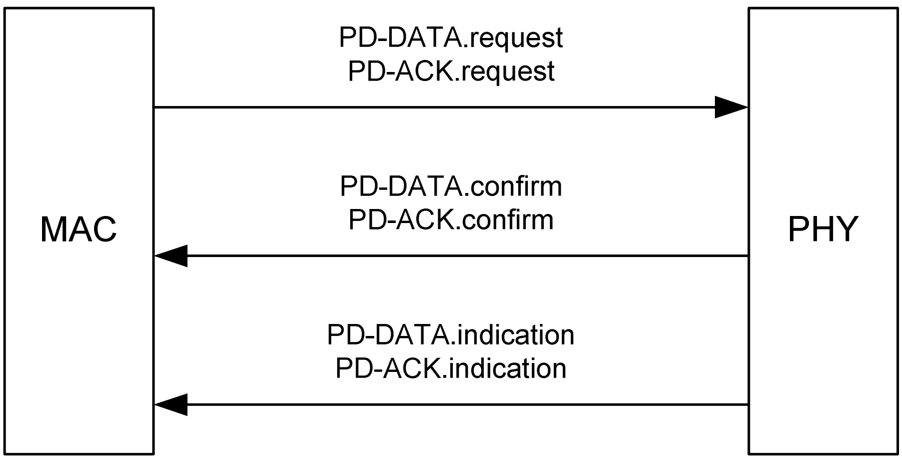


Figure 19 – Data transmission flow (MAC → PHY)

6.11.1.1 PD-DATA.request

The PD-DATA.request primitive is generated by a local MAC sublayer entity and issued to its PHY entity to request the transmission of an MPDU.

The semantics of the PD-DATA.request primitive is as follows:

```
PD-DATA.request (
                psduLength
                psdu
                )
```

Table 15 specifies the parameters for the PD-DATA.request primitive.

Table 15 – PD-DATA.request primitive

Name	Type	Valid Range	Description
psduLength	Integer	0 to 400	The number of bytes contained in the PSDU to be transmitted by the PHY entity.
psdu	Bytes	------	The set of bytes forming the PSDU request to transmit by the PHY entity.

6.11.1.2 PD-DATA.confirm

The PD-DATA.confirm primitive confirms the end of the transmission of an MPDU (i.e., PSDU) from a local PHY entity to a peer PHY entity.

The semantics of the PD-DATA.confirm primitive is as follows:

```
PD-DATA.confirm (
                    status
                    )
```

Table 16 specifies the parameters for the PD-DATA.confirm primitive.

Table 16 – PD-DATA.confirm primitive

Name	Type	Valid Range	Description
Status	Enumeration	SUCCESS (0), FAILED (1)	The result of the request to transmit a packet.

6.11.1.3 PD-DATA.indication

The PD-DATA.indication primitive indicates the transfer of an MPDU (i.e., PSDU) from the PHY to the local MAC sublayer entity.

The semantics of the PD-DATA.indication primitive is as follows:

```
PD-DATA.indication (
                        psduLength,
                        psdu,
                        ppduLinkQuality
                        )
```

Table 17 specifies the parameters for the PD-DATA.indication primitive.

Table 17 – PD-DATA.indication primitive

Name	Type	Valid Range	Description
psduLength	Integer	Maximum 400	The number of bytes contained in the PSDU received by the PHY entity.
Psdu	Bytes	-------	The set of bytes forming the PSDU received by the PHY entity.
ppduLinkQuality	Integer	0 to 255	Link Quality Indicator (LQI) value measured during reception of the PPDU

6.11.1.4 PD-ACK.request

The PD-ACK.request primitive demands the MAC sublayer entity send an ACK frame to the PHY entity.The semantics of the PD-ACK.request primitive is as follows:

```
PD-ACK.request (
```

FCH

)

Table 18 specifies the parameter for the PD-ACK.request primitive.

Table 18 – PD-ACK.request primitive

Name	Type	Valid Range	Description
FCH	Structure	See 6.6.5	MAC layer provides all Frame Control Header parameters to construct FCH frame for ACK.

6.11.1.5 PD-ACK.confirm

The PD-ACK.confirm confirms the end of the transmission of an ACK packet.

The semantics of the PD-ACK.confirm primitive is as follows:

PD-ACK.confirm (

Status

)

Table 19 specifies the parameter for the PD-ACK.confirm primitive.

Table 19 – PD-ACK.confirm primitive

Name	Type	Valid Range	Description
Status	Enumeration	SUCCESS (0), TXBUSY (1)	Confirm transmission of ACK frame.

6.11.1.6 PD-ACK.indication

The PD-ACK.indication primitive indicates the MAC entity reception of ACK frame from the PHY.

The semantics of the PD-ACK.indication primitive is as follows:

PD-DATA.indication (

FCH

)

Table 20 specifies the parameter for the PD-ACK.indication primitive.

Table 20 – PD-ACK.indication primitive

Name	Type	Valid Range	Description
FCH	Structure	See 6.6.5	MAC layer receives all Frame Control Header parameters from PHY layer.

6.11.2 Management primitives

6.11.2.1 General

This clause describes the different management primitives accessible between the MAC and PHY layers.

As shown in Figure 20, there are three types of management primitives, which are Get, Set and Confirm used to initiate commands or retrieve data from Phy :

- PLME-SET.request function configures PHY to initial specific function,

- PLME-GET.request to retrieve specific parameters from PHY,
- PLME-xxx.confirm reports the result of an action initiated by MAC.

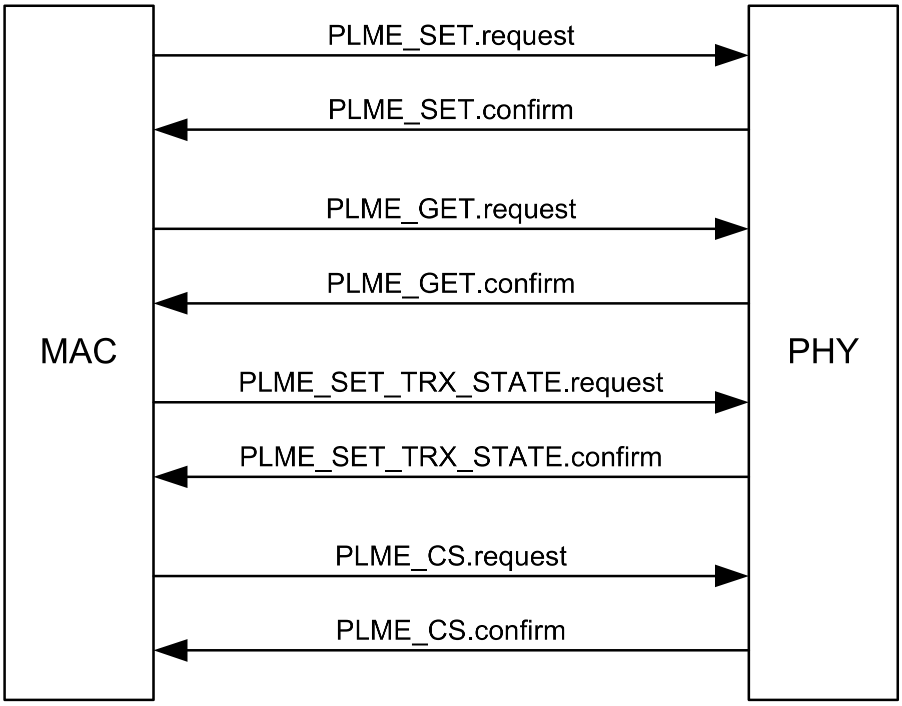


Figure 20 – Management Primitive flow

6.11.2.2 PLME_SET.request

The semantics of the PLME-SET.request primitive is as follows:

```
PLME_SET.request  (
                    TXPower,
                    AGCGain,
                    ModulationType,
                    ToneMap,
                    PreEmphasis,
                    ToneMask
                    )
```

Table 21 specifies the parameters for the PLME-SET.request primitive.

Table 21 – PLME-SET.request primitive

Name	Type	Valid Range	Description
TXPower	Integer	0x00–0x20	MAC layer uses this primitive to notify PHY about the gain/power setting PHY has to use to transmit the next packet.
AGCGain	Integer	0x0–0x3F	MAC changes the AGC gain to a desired energy level.
ModulationType	Integer	0x0-0x2	Set the TX modulation scheme for the next transmitted frame.
ToneMap	Bitmap	-----	Tone map parameter. The value of 0 indicates to the remote transmitter that

Name	Type	Valid Range	Description
			dummy data should be transmitted on the corresponding sub-carrier while a value of 1 indicates that valid data should be transmitted on the corresponding sub-carrier.
PreEmphasis	Integer	0x00-0x1F	Specify transmit gain for each 10khz section of the available spectrum.
ToneMask	Bitmap	----	Tone Mask parameter. The value of 0 indicates tone is notched while, the value 1 indicates that tone is enabled.

6.11.2.3 PLME_SET.confirm

PHY stores new parameters and returns new stored value back to MAC layer.

The semantics of the PLME-SET.confirm primitive is as follows:

PLME_SET.confirm (
TXPower,
AGCGain,
ModulationType,
ToneMap,
PreEmphasis,
ToneMask
)

Table 22 specifies the parameters for the PLME_SET.confirm primitive.

Table 22 – PLME-SET.confirm primitive

Name	Type	Valid Range	Description
TXPower	Integer	0x00–0x20	Returns new stored value back to MAC layer
AGCGain	Integer	0x00–0x3F	Returns new stored value back to MAC layer (optional)
ModulationType	Integer	0x0-0x2	Returns new stored value back to MAC layer
ToneMap	Bitmap	-----	Returns new stored value back to MAC layer
PreEmphasis	Integer	0x00-0x1F	Returns new stored value back to MAC layer
ToneMask	Bitmap	-----	Returns new stored value back to MAC layer

6.11.2.4 PLME_GET.request

The PLME-GET.request primitive requests PHY to get the parameters described in Table 58.

The semantics of the PLME-GET.request primitive is as follows:

PLME_GET.request ( )

6.11.2.5 PLME_GET.confirm

The semantics of the PLME-GET.confirm primitive is as follows:

PLME_SET.confirm (
SNR
CarrierSNR
RXSensitivity
ZCTDifferential

)

Table 23 specifies the parameters for the PLME-GET.confirm primitive.

Table 23 – PLME-GET.confirm primitive

Name	Type	Valid Range	Description
SNR	Integer	0x00–0x1F	Channel SNR value in dB.
CarrierSNR	Array	0x00–0x1F for each carriers	SNR value per each carrier (in dB)
RX Sensitivity	Integer	0x0-0x1F	Receiver sensitivity (in dB)
ZCTDifferential	Integer	0x00–0xFF	PHY computes and provide time difference between local 50 Hz phase and remote end.

6.11.2.6 PLME_SET_TRX_STATE.request

The PLME_SET_TRX_STATE.request primitive requests PHY to change the state of the TX / RX.

The semantics of the PLME_SET_TRX_STATE.request primitive is as follows:

PLME_SET.TRX_STATE.request(
State
)

Table 24 specifies the parameters for the PLME_SET.TRX_STATE.request primitive.

Table 24 – PLME_SET.TRX_STATE.request primitive

Name	Type	Valid Range	Description
State	Enumeration	TXON_RXOFF	Transmission state: turns off the RX PHY when transmitting packets.
		TXOFF_RXON	Receive state: turns off the transmitter and enable RX to wait packets.

6.11.2.7 PLME_SET_TRX_STATE.confirm

The PLME_SET_TRX_STATE.confirm primitive confirms the changing PHY state.

The semantics of the PLME_SET_TRX_STATE.confirm primitive is as follows:

PLME_SET_TRX_STATE.confirm(
Status
)

Table 25 specifies the parameters for PLME_SET_TRX_STATE.confirm primitive.

Table 25 – PLME_SET.TRX_STATE.confirm primitive

Name	Type	Valid Range	Description
Status	Enumeration	SUCCESS, TXBUSY, RXBUSY	Confirm RX and TX are set or provide error message if TX or RX are busy.

6.11.2.8 PLME_CS.request

The PLME-CS.request primitive requests PHY to get media status using carrier sense.

The semantics of the PLME_CS.request primitive is as follows:

PLME_CS.request ()

6.11.2.9 PLME_CS.confirm

The PLME-CS.confirm primitive reports media status.

The semantics of the PLME_CS.confirm primitive is as follows:

PLME_CS.confirm (
Status
)

Table 26 specifies the parameters for the PLME_CS.confirm primitive.

Table 26 – PLME_CS.confirm primitive

Name	Type	Valid Range	Description
Status	Enumeration	IDLE, BUSY	Powerline media status

7 Data link layer specification

7.1 Introduction

The PLC OFDM Type 2 data link layer specification comprises two sublayers:

– The MAC sublayer based on IEEE 802.15.4; and

– The Adaptation sublayer based on RFC 4944: Transmission of IPv6 Packets over IEEE 802.15.4 Networks (6LowPan).

The present standard specifies the necessary selections from and extensions to these standards.

7.2 Conventions

In the present section, the status of each requirement from the reference documents is given using the following convention:

– I = "Informative". The statements of the reference document are provided for information only;

– N = “Normative”: The statements of the reference document apply without modifications or remarks;

– S = “Selection”: The statements of the reference document apply with the selections specified;

– E = “Extension”: The statements of the reference document apply with the extensions specified;

– N/R = “Not Relevant”: The statements of the reference document do not apply. An explanation may be given under the part title.

7.3 MAC sublayer specification

7.3.1 MAC sublayer service specification (based on IEEE 802.15.4 clause 7.1)

7.3.1.1 Selections from IEEE 802.15.4 clause 7.1: MAC sublayer service specification

The MAC sublayer service specification as described in clause 7.1 of IEEE 802.15.4-2006 applies, with the selections specified in Table 27.

Table 27 – Selections from IEEE 802.15.4 clause 7.1

Clause	Title & remarks/modifications	Statement
7.1	MAC sublayer service specification	N
7.1.1	MAC data service - MCPS-PURGE primitives are not used in this specification.	S
7.1.1.1	MCPS-DATA.request	N
7.1.1.1.1	Semantics of the service primitive - Extension: Additional QualityOfService parameter: see 7.3.1.2. - Only non beacon-enabled PAN is used; - Bit b2 of TxOptions parameter must always be 0 See Annex E of the present document for complete semantics description of this primitive.	S, E
7.1.1.1.2	Appropriate usage	N
7.1.1.1.3	Effect on receipt - GTS transmission is not used; - Only unslotted CSMA-CA for nonbeacon-enabled PAN is used; - Indirect transmission is not supported	S
7.1.1.2	MCPS-DATA.confirm	N
7.1.1.2.1	Semantics of the service primitive	N
7.1.1.2.2	When generated	N
7.1.1.2.3	Appropriate usage	N
7.1.1.3	MCPS-DATA.indication	N
7.1.1.3.1	Semantics of the service primitive - Extension: Additional QualityOfService parameter: see clause 7.3.1.2. See Annex E of the present document for complete semantics description of this primitive.	S, E
7.1.1.3.2	When generated	N
7.1.1.3.3	Appropriate usage	N
7.1.1.4	MCPS-PURGE.request - MCSP-PURGE.request is not handled in the present specification.	N/R
7.1.1.5	MCPS-PURGE.confirm - MCSP-PURGE.confirm is not handled in the present specification.	N/R
7.1.1.6	Data service message sequence chart	N
7.1.2	MAC management service	N
7.1.3	Association primitives	N/R
7.1.3.1	MLME-ASSOCIATE.request - MLME-ASSOCIATE.request is not used in this specification. Association is performed by the 6LoWPAN Bootstrap Protocol described in clause 7.4.5 of the present document.	N/R
7.1.3.2	MLME-ASSOCIATE.indication - MLME-ASSOCIATE.indication is not used in this specification. Association is performed by the 6LoWPAN Bootstrap Protocol described in clause 7.4.5 of the present document.	N/R
7.1.3.3	MLME-ASSOCIATE.response - MLME-ASSOCIATE.response is not used in this specification. Association is performed by the 6LoWPAN Bootstrap Protocol described in clause 7.4.5 of the present document.	N/R
7.1.3.4	MLME-ASSOCIATE.confirm - MLME-ASSOCIATE.confirm is not used in this specification. Association is	N/R

Clause	Title & remarks/modifications	Statement
	performed by the 6LoWPAN Bootstrap Protocol described in clause 7.4.5 of the present document.	
7.1.3.5	Association message sequence chart - The association message sequence chart described in figure 31 must be ignored for this specification, as association is performed using the bootstrap mechanism described in clause 7.4.5 of the present document.	N/R
7.1.4	Disassociation primitive	N/R
7.1.4.1	MLME-DISASSOCIATE.request - MLME-DISASSOCIATE.request is not used in this specification. Disassociation is performed by the 6LoWPAN Bootstrap Protocol described in clause 7.4.5 of the present document.	N/R
7.1.4.2	MLME-DISASSOCIATE.indication - MLME-DISASSOCIATE.indication is not used in this specification. Disassociation is performed by the 6LoWPAN Bootstrap Protocol described in clause 7.4.5 of the present document.	N/R
7.1.4.3	MLME-DISASSOCIATE.confirm - MLME-DISASSOCIATE.confirm is not used in this specification. Disassociation is performed by the 6LoWPAN Bootstrap Protocol described in clause 7.4.5 of the present document.	N/R
7.1.4.4	Disassociation message sequence chart - The disassociation message sequence chart described in figure 31 must be ignored for this specification, as disassociation is performed using the bootstrap mechanism described in clause 7.4.5 of the present document.	N/R
7.1.5	Beacon notification primitive	N/R
7.1.5.1	MLME-BEACON-NOTIFY.indication - Only nonbeacon-enabled PANs are used. - This primitive is generated upon reception of a beacon during an active scan	S
7.1.6	Primitives for reading PIB attributes	N
7.1.6.1	MLME-GET.request	N
7.1.6.1.1	Semantics of the service primitive	N
7.1.6.1.2	Appropriate usage	N
7.1.6.1.3	Effect on receipt	N
7.1.6.2	MLME-GET.confirm	N
7.1.6.2.1	Semantics of the service primitive	N
7.1.6.2.2	When generated	N
7.1.6.2.3	Appropriate usage	N
7.1.7	GTS management primitives - GTS are not used in the present specification	N/R
7.1.8	Primitives for orphan notification - Beacon synchronization is not used in the present specification	N/R
7.1.9	Primitives for resetting the MAC sublayer	N
7.1.9.1	MLME-RESET.request	N
7.1.9.1.1	Semantics of the service primitive	N
7.1.9.1.2	Appropriate usage	N
7.1.9.1.3	Effect on receipt	N
7.1.9.2	MLME-RESET.confirm	N
7.1.9.2.1	Semantics of the service primitive	N
7.1.9.2.2	When generated	N
7.1.9.2.3	Appropriate usage	N

Clause	Title & remarks/modifications	Statement
7.1.10	Primitives for specifying the receiver enable time - The primitives for specifying the receiver enable time are not used in the present application of the norm. The receiver is always enabled	N/R
7.1.11	Primitives for channel scanning	N
7.1.11.1	MLME-SCAN.request	N
7.1.11.1.1	Semantics of the service primitive - The only supported values for the ScanType parameter is 0x01 for active scan. - The ScanChannels parameter is not used, and all of its 27 bits must be set to 0. - The ChannelPage parameter is not used and must always be set to 0. - The SecurityLevel must be 0. Thus the KeyIdMode, KeyIndex and KeySource parameters can be ignored and set to 0.	S
7.1.11.1.2	Appropriate usage - Only active scan is supported - ED scans, passive scans and orphan scans are not used. All devices must be capable of performing active scans.	S
7.1.11.1.3	Effect on receipt - Only active scan is supported, - ED scan, passive scan and orphan scan are not supported, - There is no physical channel notion during the scans, as the underlying PHY layer does not support multiple channels.	S
7.1.11.2	MLME-SCAN.confirm	N
7.1.11.2.1	Semantics of the service primitive - The only supported values for the ScanType parameter is 0x01 for active scan. - The UnscannedChannels parameter is not used, and all of its 27 bits must be set to 0. - The ChannelPage parameter is not used and must always be set to 0. - The EnergyDetectList parameter is not used, and must always be null.	S
7.1.11.2.2	When generated - Only active scan is supported - ED scan, passive scan and orphan scan are not supported.	S
7.1.11.2.3	Appropriate usage	N
7.1.11.3	Channel scan message sequence chart - Figure 79 must be ignored (ED scan not supported) - Figure 82 must be ignored (passive scan not supported) - Figure 86 must be ignored (orphan scan not supported) - Active scan message sequence chart is specified in clause 7.4.5.2.2 of the present document, and replaces figure 83 of the reference document.	S
7.1.12	Communication status primitive	N
7.1.12.1	MLME-COMM-STATUS.indication	N
7.1.12.1.1	Semantics of the service primitive - Valid values for the status parameters are: SUCCESS, CHANNEL_ACCESS_FAILURE, NO_ACK, COUNTER_ERROR, FRAME_TOO_LONG, IMPROPER_KEY_TYPE, IMPROPER_SECURITY_LEVEL, SECURITY_ERROR, UNAVAILABLE_KEY, UNSUPPORTED_LEGACY, UNSUPPORTED_SECURITY or INVALID_PARAMETER	S
7.1.12.1.2	When generated - This primitive is not used to notify the upper layer about association, disassociation, indirect transmission and transactions management	S
7.1.12.1.3	Appropriate usage	N

Clause	Title & remarks/modifications	Statement
7.1.13	Primitives for writing PIB attributes	N
7.1.13.1	MLME-SET.request	N
7.1.13.1.1	Semantics of the service primitive	N
7.1.13.1.2	Appropriate usage	N
7.1.13.1.3	Effect on receipt	N
7.1.13.2	MLME-SET.confirm	N
7.1.13.2.1	Semantics of the service primitive	N
7.1.13.2.2	When generated	N
7.1.13.2.3	Appropriate usage	N
7.1.14	Primitives for updating the superframe configuration - This primitive is only used on the PAN coordinator in case of network formation (see clause 7.5.1 of the present document).	S
7.1.14.1	MLME-START.request - This primitive is only used to initiate a new PAN.	S
7.1.14.1.1	Semantics of the service primitive - Primitive parameters must be set as described in clause 7.5.1 of the present document.	S
7.1.14.1.2	Appropriate usage	N
7.1.14.1.3	Effect on receipt - Primitive parameters must be set as described in clause 7.5.1 of the present document.	S
7.1.14.2	MLME-START.confirm	N
7.1.14.2.1	Semantics of the service primitive	N
7.1.14.2.2	When generated	N
7.1.14.2.3	Appropriate usage	N
7.1.14.3	Message sequence chart for updating the superframe configuration - Figure 38 must be ignored.	N/R
7.1.15	Primitives for synchronizing with a coordinator - This part is used to inform the upper layers in case of a PAN ID conflict or PAN realignment.	S
7.1.15.1	MLME-SYNC.request	N/R
7.1.15.2	MLME-SYNC-LOSS.indication - PAN ID conflict detection is performed by the 6LoWPAN Bootstrap Protocol as described in clause 7.5.2 of the present document.	N/R
7.1.15.3	Message sequence chart for synchronizing with a coordinator - Synchronization with beacons is not used in the present specification	N/R
7.1.16	Primitives for requesting data from a coordinator - Indirect transmission and transactions are not supported by the present specification	N/R
7.1.17	MAC enumeration description	N

7.3.1.2 Extensions to IEEE 802.15.4 clause 7.1: additional QualityOfService parameter

As shown in Table 28, the Quality of Service (QOS) parameter defines the level of priority assigned to the MSDU to be transmitted. The Annex D defines the priority mechanism of the PLC OFDM Type 2.

Table 28 – QualityOfService parameter definition

Name	Type	Valid range	Description
QualityOfService	Integer	0x00 – 0x02	The QOS (Quality of Service) parameter of the MSDU to be transmitted by the MAC sublayer entity. This value can take one of the following values: 0 = Normal priority, 1 = High priority, 2 = Contention free

7.3.2 MAC frame formats (based on IEEE 802.15.4 clause 7.2)

7.3.2.1 Selections from IEEE 802.15.4 clause 7.2: MAC frame formats

The MAC frame formats as described in clause 7.2 of IEEE 802.15.4-2006 apply, with the selections specified in Table 29.

Table 29 – Selections from clause 7.2 of the IEEE 802.15.4

Clause	Title & remarks/modifications	Statement
7.2	MAC frame formats	N
7.2.1	General MAC frame format - Segment Control fields are added to MHR (see 7.3.2.2) - Detailed description of the Segment Control fields is shown in Table 31.	E
7.2.1.1	Frame control field	N
7.2.1.1.1	Frame type subfield - The present specification does not use acknowledgement frame type value. - The detailed ACK implementation is described in clause 5 and annex Annex F of the present document. An acknowledgment can be sent by invoking the PD-ACK.request primitive (see 6.11.1.1).	S
7.2.1.1.2	Security Enabled subfield	N
7.2.1.1.3	Frame Pending subfield - Indirect transmission is not supported, so this bit must always be set to 0.	S
7.2.1.1.4	Acknowledgement Request subfield - The present specification translates the Acknowledgment Request subfield to the proper delimiter type of frame control header. - The detailed ACK implementation is described in clause 5 and annex Annex F of the present document. An acknowledgement can be sent by invoking the PD-ACK.request primitive (see 6.11.1.1).	S
7.2.1.1.5	PAN ID compression subfield	N
7.2.1.1.6	Destination Addressing Mode subfield	N
7.2.1.1.7	Frame Version subfield - These 2 bits are reserved for future use. In this version of the specification they must be set to 0.	S
7.2.1.1.8	Source Addressing Mode subfield	N
7.2.1.2	Sequence Number field	N
7.2.1.3	Destination PAN Identifier field	N
7.2.1.4	Destination Address field	N
7.2.1.5	Source PAN Identifier field	N
7.2.1.6	Source Address field	N
7.2.1.7	Auxiliary Security Header field - Possible lengths for the Auxiliary Security Header are 0 and 5 bytes (see clause 8)	S

Clause	Title & remarks/modifications	Statement
7.2.1.8	Frame Payload field	N
7.2.1.9	FCS field	N
7.2.2	Format of individual frame types	N
7.2.2.1	Beacon frame format	N
7.2.2.1.1	Beacon frame MHR fields	N
7.2.2.1.2	Superframe Specification field - Beacons are not transmitted at regular time intervals (beaconless network). Therefore the Beacon Order parameter of the Superframe Specification field is not used and must always be set to 0. - The receiver is active all the time when not transmitting. Therefore the Superframe Order parameter of the Superframe Specification field is not used and must always be set to 0. - No superframe structure is used for communication, so the Final CAP Slot parameter of the Superframe Specification field is not used and must always be set to 0. - Devices will not be operating on batteries, so the Battery Life Extension subfield of the Superframe Specification field is not used and must always be set to 0. - Within the framework of the present standard, the association is performed by the 6LoWPAN Bootstrap Protocol in the upper layer, so the Association Permit parameter of the Superframe Specification field is meaningless here, and should always be set to 1. If another profile is used, this field should be set as described in clause 7.2.2.1.2 of IEEE 802.15.4.	S
7.2.2.1.3	GTS Specification field - The GTS Descriptor Count must always be set to 0 (GTS are not supported). - The PAN coordinator never accepts GTS request, therefore the GTS Permit parameter of the GTS Specification field must always be set to 0.	S
7.2.2.1.4	GTS Direction field - The GTS feature is not used, and the GTS Direction field must not be present in the frame	N/R
7.2.2.1.5	GTS List field - The GTS feature is not used, and considering the values of the GTS specification field described in clause 7.2.2.1.3 of the IEEE 802.15.4, this list must be empty	N/R
7.2.2.1.6	Pending Address specification field - Indirect transmission is not supported in this specification. Consequently, the Number of Short Addresses Pending is always 0, and Number of Extended Addresses Pending is also 0.	S
7.2.2.1.7	Address List field - Indirect transmission is not used, and this field must not be present in beacons.	N/R
7.2.2.1.8	Beacon Payload field - In the current version of this specification, the beacon payload field is empty.	S
7.2.2.2	Data frame format	N
7.2.2.2.1	Data frame MHR fields	N
7.2.2.2.2	Data payload field	N
7.2.2.3	Acknowledgement frame format - Acknowledgement frame format described in clause 7.2.2.3 of IEEE 802.15.4 is not relevant. - The detailed ACK implementation is described in clause 5 and annex Annex F of the present document. An acknowledgement can be sent by invoking the PD-ACK.request primitive (see 6.11.1.1).	S
7.2.2.4	MAC command frame format	N
7.2.2.4.1	MAC command frame MHR fields	N

Clause	Title & remarks/modifications	Statement
7.2.2.4.2	Command Frame Identifier field	N
7.2.2.4.3	Command Payload field	N
7.2.3	Frame compatibility - The use of the Frame Version subfield is reserved.	N/R

7.3.2.2 Extensions to IEEE 802.15.4 clause 7.2: MAC frame formats

Table 30 and Table 31 define the Segment Control field added in the MAC Header (MHR) specified in IEEE 802.15.4 clause 7.2.

Table 30 – General MAC frame format

Octets: 3	2	1	0/2	0/2/8	0/2	0/2/8	0/5/6/10	Variable	2
Segment Control	Frame Control	Sequence Number	Destination PAN	Destination Address	Source PAN	Source Address	Auxiliary Security Header	Frame payload	FCS
MHR								MAC payload	MFR

Table 31 – Segment control fields

Field	Byte	Bit number	Bits	Definition
RES	0	7-4	4	Reserved
TMR	0	3	1	Tone map request 1: Tone map is requested 0: Tone map is not requested
CC	0	2	1	Contention Control: 0: contention is allowed in next contention state 1: contention free access
CAP	0	1	1	Channel access priority: 0: Normal 1: High
LSF	0	0	1	Last Segment Flag 0: Not last segment 1: Last segment
SC	1	7-2	6	Segment Count
SL[9-8]	1	1-0	2	Segment Length of MAC frame
SL[7-0]	2	7-0	8	Segment Length of MAC frame

7.3.3 MAC command frames (based on IEEE 802.15.4 clause 7.3)

7.3.3.1 Selections from IEEE 802.15.4 clause 7.3: MAC command frames

The MAC frame formats as described in clause 7.3 of IEEE 802.15.4-2006 apply, with the selections specified in Table 32.

Table 32 – Selections from clause 7.3 of the IEEE 802.15.4

Clause	Title & remarks/modifications	Statement
7.3	MAC command frames - All devices are Full Function Devices - The supported command list is defined in 7.3.3.2.1	S, E

Clause	Title & remarks/modifications	Statement
7.3.1	Association request command - Within the framework of the present standard, association is performed by the 6LoWPAN Bootstrap protocol described in clause 7.4.5.2.2 of the present document, so the clause 7.3.1 of IEEE 802.15.4 is not relevant.	N/R
7.3.2	Association response command - Within the framework of the present standard, association is performed by the 6LoWPAN Bootstrap protocol described in clause 7.4.5.2.2 of the present document, so the clause 7.3.2 of IEEE 802.15.4 is not relevant.	N/R
7.3.3	Disassociation Notification command - Within the framework of the present standard, association is performed by the 6LoWPAN Bootstrap protocol described in clause 7.4.5.2.2 of the present document, so the clause 7.3.2 of IEEE 802.15.4 is not relevant.	N/R
7.3.4	Data Request command	N/R
7.3.5	PAN ID Conflict Notification command - PAN ID Conflict Notification is performed by the adaptation layer, see 7.5.2.	N/R
7.3.6	Orphan notification command - Orphan notification is not used in the present specification	N/R
7.3.7	Beacon request command - This command must be implemented in every device	S
7.3.8	Coordinator realignment command - The coordinator realignment command is not used in the present notification	N/R
7.3.9	GTS request command - GTS are not used in the present specification	N/R

7.3.3.2 Extensions to IEEE 802.15.4 clause 7.3: MAC command frames

7.3.3.2.1 MAC command frames supported

The present standard supports the MAC command frames described in Table 33:

Table 33 – MAC command frames supported

Command frame identifier	Command name	Sub-clause
0x07	Beacon request	See clause 7.3.7 of IEEE 802.15.4
0x0a	Tone map response	See clause 7.3.3.2.2

7.3.3.2.2 The Tone Map Response

The MAC sublayer generates Tone Map Response command if Tone Map Request (TMR) bit of received packet Segment Control field is set. It means that a packet originator requested tone map information from destination device. The destination device has to estimate this particular communication link between two points and choose optimal PHY parameters. The tone map information includes the index associated with PHY parameters: number of used tones and allocation (Tone Map), modulation mode and TX power control parameters. The *Tone Map Response* message parameters are described in table below.

Figure 21 shows the format of the *Tone Map Response* message that is sent from the receiver back to the transmitter requesting the channel estimation.

Preamble	Frame Control Field	Tone map Response
N Symbols	39 bits	M bytes

Figure 21 – Frame structure of a Tone Map Response message

The channel estimation response command frame must be formatted as illustrated in Table 34:

Table 34 – Tone map response format

Octets: (see clause 7.2.2.4 of IEEE 802.15.4)	1	7 (for CENELEC)
MHR fields	Command frame identifier (see Table 35)	Tone Map response payload (see Table 35)

The Tone Map Response message parameters are shown in Table 35.

Table 35 – Tone Map Response message description for CENELEC A band

Field	Byte	Bit number	Bits	Definition
TXRES	0	7	1	Tx Gain resolution corresponding to one gain step. 0 : 6 dB 1 : 3 dB
TXGAIN	0	6-3	4	Desired Transmitter gain specifying how many gain steps are requested.
MOD	0	2-1	2	Modulation type: 0 – Robust; 1 – DBPSK 2 – DQPSK 3 – D8PSK
TM[8]	0	0	1	Tone Map [8]
TM[0:7]	1	7-0	8	Tone Map [0:7]
LQI	2	7-0	8	Link Quality Indicator
TXCOEF[3:0]	3	7-4	4	Specifies number of gain steps requested for 10kHz-20kHz spectrum (optional)
TXCOEF[7:4]	3	3-0	4	Specifies number of gain steps requested for 20kHz-30kHz spectrum (optional)
TXCOEF[11:8]	4	7-4	4	Specifies number of gain steps requested for 30kHz-40kHz spectrum (optional)
TXCOEF[15:12]	4	3-0	4	Specifies number of gain steps requested for 40kHz-50kHz spectrum (optional)
TXCOEF[19:16]	5	7-4	4	Specifies number of gain steps requested for 50kHz-60kHz spectrum (optional)
TXCOEF[23:20]	5	3-0	4	Specifies number of gain steps requested for 60kHz-70kHz spectrum (optional)
TXCOEF[27:24]	6	7-4	4	Specifies number of gain steps requested for 70kHz-80kHz spectrum (optional)
TXCOEF[31:28]	6	3-0	4	Specifies number of gain steps requested for 80kHz-90kHz spectrum (optional)

Where:

– MOD: Parameter that specifies the desired modulation type. The receiver computes the SNR of the *Tone Map Request* message that it receives from the transmitter and it decides which of the three modulation modes (DBPSK, DQPSK, D8PSK or Robust) it wants the transmitter to use when sending next data frame or *Tone Map Request* message. Table 36 lists the allowed bit values and the modulation modes they correspond to;

Table 36 – Modulation Method Field

MOD Value	Interpretation
00	Robust Modulation
01	DBPSK Modulation
10	DQPSK Modulation
11	D8PSK Modulation

– TXRES: Parameter that specifies the transmit gain resolution corresponding to one gain step;

– TXGAIN: Parameter that specifies to the transmitter the total amount of gain that it should apply to its transmitted signal. The value in this parameter shall specify the total number of gain steps needed. The receiver computes the received signal level and compares it to a VTARGET (pre-defined desired receive level). The difference in dB between the two values is mapped to a 5-bit value that specifies the amount of gain increase or decrease that the transmitter shall apply to the next frame to be transmitted. A “0” in the most significant bit indicates a positive gain value, hence an increase in the transmitter gain and a 1 indicates a negative gain value, hence a decrease in the transmitter gains. A value of TXGAIN = 0 informs the transmitter to use the same gain value it used for previous frame (Default value);

– TM: Parameter that specifies the Tone Map. The receiver estimates the per-tone quality of the channel and maps each sub-band (6 tones per sub-band) to a one-bit value where a value of 0 indicates to the remote transmitter that dummy data should be transmitted on the corresponding sub-carrier while a value of “1” indicates that valid data should be transmitted on the corresponding sub-carrier;

– TXCOEF (optional): Parameter that specifies transmitter gain for each 10 kHz section of the available spectrum. The receiver measures the frequency-dependent attenuation of the channel and may request the transmitter to compensate for this attenuation by increasing the transmit power on sections of the spectrum that are experiencing attenuation in order to equalize the received signal. Each 10 kHz section is mapped to a 4-bit value where a “0” in the most significant bit indicates a positive gain value, hence an increase in the transmitter gain is requested for that section, and a “1” indicates a negative gain value, hence a decrease in the transmitter gain is requested for that section. Implementing this feature is optional and it is intended for frequency selective channels. If this feature is not implemented, the value zero should be used.

On reception of a Tone Map Response command frame, the MAC sublayer updates the neighbour table with the corresponding Tone Map and communication parameters for that device. If no entry already exists in the table for that device a new entry should be added, based on implementation dependant limitations. The neighbour table is defined in Table 42.

The following procedure must be used to perform the adaptive tone mapping function:

a) When a station is ready to transmit data it will first check if the neighbour table already has a record related to the destination device address. If the record don’t exists or aged (Age counter is 0), the MAC sublayer sets the TMR bit of outgoing packet Segment Control field and requests new Tone Map information. In this case the MAC data should be sent in Robust mode;

b) If a neighbour table record exists and it is not aged the MAC sublayer does not need to send Tone Map Request message. In this case, the MAC sublayer uses information from the neighbour table to properly configure physical TX in transmitting mode and construct Frame Control Header (FCH) of the outgoing frame;

c) When the destination station receives a data frame it shall check the Tone Map Request bit in the Segment Control field. If the bit is set, the destination station must measure the per-carrier quality of the channel, construct and send a Tone Map Response message back to the originator station. The destination station must not send a Tone Map Response message if the Tone Map Request bit is not set. The Tone Map Response message shall always be transmitted using default Robust modulation. The destination device uses parameters from the Frame Control Header to decode the MAC data fields;

d) The destination station must attempt to send a Tone Map Response message as soon as possible after receiving a Tone Map Request message from the source station;

e) If the source station receives a Tone Map Response message, it will update a neighbour table record related to the destination address with new Tone Map, modulation and TX gain parameters. If the record doesn’t exist, the MAC sublayer will create a new one. The Age counter should be set to defined value (see clause 7.3.4). After receiving a Tone Map Response message, a device shall begin to use the updated neighbour table information for all transmissions to the associated destination until the Age counter will reach the value “0”;

f) If the source station does not receive a Tone Map Response message after transmitting a Tone Map Request message to a certain destination, it must set the Tone Map Request bit in the Segment Control of the next MAC data frame that it wants to transmit to the same destination. In other words, the MAC sublayer will continue to transmit a Tone Map Request message to the same destination;

g) The MAC sublayer shall not send a Tone Map Request message to the destination device if no data sent to this device.

The Tone Map request/response message sequence chart is shown in 7.3.7.2.4.

7.3.4 MAC constants and PIB attributes (based on IEEE 802.15.4 clause 7.4)

7.3.4.1 Selections from IEEE 802.15.4 clause 7.4: MAC constants and PIB attributes

The MAC frame formats as described in clause 7.4 of IEEE 802.15.4 apply, with the selections specified in Table 37.

Table 37 – Selections from clause 7.4 of the 802.15.4

Clause	Title & remarks/modifications	Statement
7.4	MAC constants and PIB attributes	N
7.4.1	MAC constants - The aBaseSlotDuration parameter is not used and must be set to 0. - The aBaseSuperframeDuration parameter is not used and must be set to 0. - The aExtendedAddress parameter must be equal to the EUI-48 address of the device mapped to a EUI-64 address. - The aGTSDescPersistenceTime parameter is not used and must be set to 0. - The aMaxBeaconOverhead parameter must be set to 0. - The aMaxBeaconPayloadLength parameter is not used and must be set to. - The aMaxLostBeacons parameter is not used and must be set to 0. - The aMaxMACSafePayloadSize parameter is not used and must be set to 0. - The aMaxMACPayloadSize parameter is fixed to 400 bytes by the present standard. - The aMaxMPDUUnsecuredOverhead parameter must be set to 25 (bytes). - The aMaxSIFSFrameSize parameter is not used and must be set to 0. - The aMinCAPLength parameter is not used and must be set to 0. - The aMinMPDUOverhead parameter is fixed to 9 bytes by the present standard. - The aNumSuperframeSlots parameter is not used and must be set to 0. - The aUnitBackoffPeriod parameter must be set to aSlotTime. - Extensions: Additional MAC sublayer constants are defined in 7.3.4.2.1.	S, E

Clause	Title & remarks/modifications	Statement
7.4.2	MAC PIB attributes - The macAckWaitDuration parameter must be set according to the following formula: macAckWaitDuration = aRIFS + aAckTime + aCIFS - The macAssociatedPANCoord parameter is not used and must be set to FALSE. - The macAssociationPermit parameter must always be set to TRUE, and must be read-only. - The macAutoRequest parameter is not used and must be set to FALSE. - The macBattLifeExt parameter is not used; changing it has no effect on the behaviour of the device. Its default value must be FALSE, and must not be changed. - The macBattLifeExtPeriods parameter is not used; changing it has no effect on the behaviour of the device. Its default value must be 0, and must not be changed. - The macBeaconPayload parameter is not used; changing it has no effect on the behaviour of the device. Its default value must be NULL, and must not be changed. - The macBeaconPayloadLength parameter is not used and must be set to 0. - The macBeaconOrder parameter is not used; changing it has no effect on the behaviour of the device. Its default value must be left to 15, and must not be changed. - When the macBeaconTxTime parameter reaches 0xFFFFFF, it must not change anymore. - The macGTSPermit parameter is not used; changing it has no effect on the behaviour of the device. Its default value must be FALSE, and must not be changed. - The macMaxBE parameter is fixed to 5 by the present standard. - The macMaxCSMABackoffs default value is fixed to 8 by the present standard. - The macMaxFrameTotalWaitTime parameter is not used must be set to 0. - The macMinBE parameter is fixed to 3 by the present standard. - The macMinLIFSPeriod parameter is not used; changing it has no effect on the behaviour of the device. - The macMinSIFSPeriod parameter is not used; changing it has no effect on the behaviour of the device. - The macResponseWaitTime parameter must be set to macAckWaitDuration. - The macRxOnWhenIdle parameter must always be set to TRUE. - The macSecurityEnabled parameter must always be set to TRUE. - The macShortAddress parameter must be equal to 0xFFFF when the device does not have a short address. An associated device necessarily has a short address, so that a device cannot be in the state where it is associated but does not have a short address. - The macSuperframeOrder parameter is not used, and must be left to 15. - The macSyncSymbolOffset is not used and must be set to 0. - The macTimestampSupported parameter must be set to TRUE. - The macTransactionPersistenceTime parameter is not used and must be set to 0. - Extensions: Additional set of IB attributes are defined in 7.3.4.2.	S, E

7.3.4.2 Extensions to IEEE 802.15.4 clause 7.4: MAC constants and PIB attributes

7.3.4.2.1 Additional MAC sublayer constants to IEEE 802.15.4 clause 7.4.1

Table 38 defines the list of MAC sublayer constants added by the present standard:

Table 38 – Additional MAC sublayer constants to IEEE 802.15.4 clause 7.4.1

Constant	Description	Value
aSymbolTime	Defines the duration of one symbol on physical layer (milliseconds).	-
aSlotTime	The duration of the contention window (in symbols)	2
aCIFS	Defines the contention interframe space (number of	10

Constant	Description	Value
	symbols). It is defined in Annex D of the present document.	
aRIFS	Defines the response interframe space (number of symbols). It is defined in Annex D of the present document.	10
aEIFS	Defines the extended interframe space (number of symbols). It is defined in Annex D of the present document.	252
aMinFrameSize	Defines the minimum MAC frame size in symbols.	4
aMaxFrameSize	Defines the maximum MAC frame size in symbols.	252
aAckTime	Defines the acknowledgment maximum time in symbols.	23

7.3.4.2.2 Additional MAC sublayer attributes to IEEE 802.15.4 clause 7.4.2

Table 39 defines the list of MAC sublayer attributes added by the present standard:

Table 39 – Additional attributes to IEEE 802.15.4 clause 7.4.2

Attribute	Identifier	Type	Range	Description	Default value
macHighPrioirtyWindowSize	0x01000113	Unsigned Integer	0-7	The high priority contention window size in number of slots. Default value is 7*aSlotTime	7
macTxDataPacketCount	0x02000101	Unsigned Integer	0 – 4 294 967 295	Statistic counter of successfully transmitted MSDUs	0
macRxDataPacketCount	0x02000202	Unsigned Integer	0 – 4 294 967 295	Statistic counter of successfully received MSDUs	0
macTxCmdPacketCount	0x02000201	Unsigned Integer	0 – 4 294 967 295	Statistic counter of successfully transmitted command packets	0
macRxCmdPacketCount	0x02000102	Unsigned Integer	0 – 4 294 967 295	Statistic counter of successfully received command packets	0
macCSMAFailCount	0x02000103	Unsigned Integer	0 – 4 294 967 295	Statistic counter of failed CSMA transmit attempts	0
macCSMACollisionCount	0x02000104	Unsigned Integer	0 – 4 294 967 295	Statistic counter of collision due to channel busy or failed transmission	0
macBroadcastCount	0x02000106	Unsigned Integer	0 – 4 294 967 295	Statistic counter of the number of broadcast frames sent	0
macMulticastCount	0x02000107	Unsigned Integer	0 – 4 294 967 295	Statistic counter of the number of multicast frames sent	0
macBadCRCCount	0x02000108	Unsigned Integer	0 – 4 294 967 295	Statistic counter of the number of frames received with bad CRC	0
macMaxOrphanTimer	0x02000109	Unsigned Integer	0 – 4 294 967 295	The maximum number of seconds without communication with a particular device after which it is declared as an orphan.	0

Attribute	Identifier	Type	Range	Description	Default value
macNeighborTable	0x1B000100	Set	-	The neighbour table defined in 7.3.5.2.	-
macNumberOfHops	0x02000110	Unsigned Integer	0 – 8	The number of hops to reach the PAN coordinator.	0
macFreqNotching	0x02000111	Bool	FALSE TRUE	S-FSK 63 and 74 kHz frequency notching. Default value is FALSE (disabled)	FALSE

7.3.4.2.3 MAC sublayer attributes and their associated ID

Table 40 defines the new identifier associated to IEEE 802.15.4 MAC sublayer attributes used by the present standard:

Table 40 – MAC sublayer attributes and their associated ID

Attribute	Identifier
macAckWaitDuration	0x01000103
macAssociationPermit	0x01000102
macMaxCSMABackoffs	0x0100010B
macMinBE	0x0100010E
macShortAddress	0x01000112
macAssociatedPANCoord	0x01000104
macCoordShortAddress	0x01000107
macMaxBE	0x0100010A
macMaxFrameTotalWaitTime	0x0100010C
macResponseWaitTime	0x01000110
macSecurityEnabled	0x01000111
macPanId	0x0100010F

7.3.5 MAC functional description (based on IEEE 802.15.4 clause 7.5)

7.3.5.1 Selections from IEEE 802.15.4 clause 7.5: MAC functional description

The MAC functional description as described in clause 7.5 of IEEE 802.15.4-2006 applies, with the selections specified in Table 41.

Table 41 – Selections from clause 7.5 of the IEEE 802.15.4

Clause	Title & remarks/modifications	Statement
7.5	MAC functional description - beacon-enabled PAN and GTS are not supported - contention free access is not implemented	S
7.5.1	Channel access - See Annex D of the present document for channel access functional description.	E
7.5.1.1	Superframe structure	N/R
7.5.1.2	Incoming and outgoing frame structure	N/R
7.5.1.3	Interframe (IFS) spacing - See Annex D of the present document for Interframe spacing description.	E
7.5.1.4	CSMA-CA algorithm - See Annex D of the present document for description of CSMA-CA algorithm	E

Clause	Title & remarks/modifications	Statement
	(including priority, ARQ, segmentation and reassembly overview).	
7.5.2	Starting and maintaining PANs	N
7.5.2.1	Scanning through channels - Passive scanning is not supported - Orphan scanning is not supported - ED scanning is not supported - Active scanning is the only supported scanning mode - As there is no channel page or channel list notion at the physical level, a scan request does not care about a particular channel.	S
7.5.2.1.1	ED channel scan - ED channel scan is not supported by the present standard	N/R
7.5.2.1.2	Active channel scan - Active channel scan is only used by an un-associated device prior to starting association and by the PAN coordinator prior to starting a new network. - As there is no channel page or channel list notion at the physical level, a scan request does not care about a particular channel.	S
7.5.2.1.3	Passive channel scan - Passive channel scan is not supported by the present standard	N/R
7.5.2.1.4	Orphan channel scan - Orphan channel scan is not supported by the present standard	N/R
7.5.2.2	PAN identifier conflict resolution	N
7.5.2.2.1	Detection - PAN conflict detection is also performed by scanning all incoming PAN Id of frames received by the devices.	S
7.5.2.2.2	Resolution - On detection of a PAN identifier conflict, a device must generate a CONFLICT frame as described in 7.5.2 of the present document.	N/R
7.5.2.3	Starting and realigning a PAN	N
7.5.2.3.1	Starting a PAN - A PAN coordinator cannot lose its MAC address. It can however be changed based on criteria out of the scope of this standard, for example in case of PAN ID conflict detection.	S
7.5.2.3.2	Realigning a PAN - PAN realignment is not supported by the present specification.	N/R
7.5.2.3.3	Realignment in a PAN - PAN realignment is not supported by the present specification.	N/R
7.5.2.3.4	Updating superframe configuration and channel PIB attributes - The macBeaconOrder parameter must always be set to 15 to have a beaconless PAN. - The phyCurrentPage and phyCurrentChannel parameters are not used, and must always be set to 0.	S
7.5.2.4	Beacon generation - Only non beacon-enabled PAN are used - Beacon must be transmitted using the Robust modulation	S
7.5.2.5	Device discovery - Device discovery is done using the active scanning procedure described in 7.4.5.2.2.2, to force a coordinator to send a beacon.	E
7.5.3	Association and disassociation	N
7.5.3.1	Association	N/R

Clause	Title & remarks/modifications	Statement
	- Association is fully described in 7.4.5 of the present document.	
7.5.3.2	Disassociation - Disassociation is fully described in 7.4.5 of the present document.	N/R
7.5.4	Synchronization	N
7.5.4.1	Synchronization with beacons - Beacon synchronization is not used in the present standard.	N/R
7.5.4.2	Synchronization without beacons	N
7.5.4.3	Orphaned device realignment - Orphaned device realignment is not used in the present specification. - Orphaned device detection is performed at the application level using a timer, which is reset each time the device receives a frame with the Destination Address field of the MAC header equal to the MAC address (either short or extended) of the device. If this timer reaches its maximum value (macMaxOrphanTimer), then the device loses its short MAC address, and must begin an association procedure.	S
7.5.5	Transaction handling - Transactions are not supported in the present standard.	N/R
7.5.6	Transmission, reception and acknowledgement	N
7.5.6.1	Transmission	N
7.5.6.2	Reception and rejection	N
7.5.6.3	Extracting pending data from a coordinator	N/R
7.5.6.4	Use of acknowledgements and retransmissions	N
7.5.6.4.1	No acknowledgement - The present standard defines an acknowledgement differently. The detailed ACK implementation is described in 6.6.3 and Annex F.	E
7.5.6.4.2	Acknowledgement - The present standard defines an acknowledgement differently. The detailed ACK implementation is described in 6.6.3 and Annex F.	E
7.5.6.4.3	Retransmissions	N
7.5.6.5	Promiscuous mode	N
7.5.6.6	Transmission scenario	N
7.5.7	GTS allocation and management - GTS are not used in the present specification	N/R
7.5.8	Frame security	N
7.5.8.1	Security related MAC PIB attributes	N
7.5.8.1.1	Key table	N
7.5.8.1.2	Device table	N
7.5.8.1.3	Minimum security level table	N
7.5.8.1.4	Frame counter	N
7.5.8.1.5	Automatic request attributes	N
7.5.8.1.6	Default key source	N
7.5.8.1.7	PAN coordinator address	N
7.5.8.2	Functional description	N
7.5.8.2.1	Outgoing frame security procedure	N
7.5.8.2.2	Outgoing frame key retrieval procedure	N
7.5.8.2.3	Incoming frame security procedure	N
7.5.8.2.4	Incoming frame security material retrieval procedure	N

Clause	Title & remarks/modifications	Statement
7.5.8.2.5	KeyDescriptor lookup table	N
7.5.8.2.6	Blacklist checking procedure	N
7.5.8.2.7	DeviceDescriptor lookup procedure	N
7.5.8.2.8	Incoming security level checking procedure	N
7.5.8.2.9	Incoming key usage policy checking procedure	N

7.3.5.2 Extensions to IEEE 802.15.4 clause 7.5: Neighbour Table

Every device must maintain a Neighbour Table, which contains information about all the devices within the POS of a device. This table is actualized each time any frame is received from a neighbouring device, and each time a Tone Map Response command is received. This table must be accessible by the Adaptation, MAC sublayers and physical layer. Each entry of this table contains the fields listed in Table 42:

Table 42 – Neighbour Table

Field Name	Size/Type	Description
ToneMap	9 bits	The Tone Map parameter defines which frequency sub-band can be used for communication with the device. A bit set to 1 means that the frequency sub-band can be used, and a bit set to 0 means that frequency sub-band must not be used.
Modulation	2 bits	Defines the modulation type to use for communicating with the device. 0x00: Robust 0x01: DBPSK 0x02: DQPSK 0x03: D8PSK
TxGain	4 bits	Defines the Tx Gain to use to transmit frames to that device
TxRes	1 bit	Defines the Tx Gain resolution corresponding to one gain step. 0 : 6 dB 1 : 3 dB
TxCoeff	8 x 4 bits	The Tx gain for each 10 kHz-wide spectrum band
LQI	8 bits	Link Quality Indicator
Age	8 bits	The remaining lifetime of the device in minutes. - When the entry is created, this value must be set to the default value 0; - When it reaches 0, a Tone Map request may be issued if data is sent to this device. Upon successful reception of a tone map response, this value is set to adpMaxAgeTime.
IsNeighbour	8 bits	Indicate either the device is a neighbour or not.

If the device receives a frame whose source address field (MAC sublayer header) does not exist in the neighbour table, it must add a new entry for that device with the following default values:

- Modulation = 0 (Robust);
- ToneMap = (all bits set to 1) AND (*adpToneMask*);
- TxGain = 0b0000;
- TxCoeff = 0xFFFFFFFF;
- LQI = 0;
- Age = 0.

The Neighbour Table is available in the Information Base under the attribute *macNeighborTable* (see 7.3.4.2.2).

7.3.6 MAC security suite specifications (selections from IEEE 802.15.4 clause 7.6)

The security suite specifications as described in clause 7.6 of IEEE 802.15.4-2006 apply, with the selections specified in Table 43.

Table 43 – Selections from clause 7.6 of the IEEE 802.15.4

Clause	Title & remarks/modifications	Statement
7.6	Security suite specification	N
7.6.1	PIB security material	N
7.6.2	Auxiliary security header	N
7.6.2.1	Integer and octet representation	N
7.6.2.2	Security Control field	N
7.6.2.2.1	Security Level subfield - Two values are allowed by the present standard: 0x00 = "none", 0x05 = "ENC-MIC-32".	S
7.6.2.2.2	Key Identifier Mode subfield - One Key Identifier Mode is allowed by the present standard: 0x01 = "Key determined from the 1-octet Key Index subfield" The number of keys is limited to 4 (range = 0-3)	S
7.6.2.3	Frame Counter field	N
7.6.2.4	Key Identifier field	N
7.6.2.4.1	Key Source subfield	N/R
7.6.2.4.2	Key Index subfield	N
7.6.3	Security operations	N
7.6.3.1	Integer and octet representation	N
7.6.3.2	CCM* Nonce	N
7.6.3.3	CCM* prerequisites	N
7.6.3.3.1	Authentication field length	N
7.6.3.4	CCM* transformation data representation	N
7.6.3.4.1	Key and nonce data inputs	N
7.6.3.4.2	a data and m data - Two values are allowed by the present standard: 0x00 = "none", 0x05 = "ENC-MIC-32".	S
7.6.3.4.3	c data output - Two values are allowed by the present standard: 0x00 = "none", 0x05 = "ENC-MIC-32".	S
7.6.3.5	CCM* inverse transformation data representation	N
7.6.3.5.1	Key and nonce data inputs	N
7.6.3.5.2	c data and a data	N
7.6.3.5.3	m data output	N

7.3.7 Message Sequence Chart Illustrating MAC – PHY interaction (based on IEEE 802.15.4 clause 7.7)

7.3.7.1 Selections from IEEE 802.15.4 clause 7.7: Message Sequence Chart Illustrating MAC

The message Sequence Chart Illustrating MAC – PHY interaction as described in clause 7.7 of IEEE 802.15.4-2006 apply, with the selections specified in Table 44.

Table 44 – Selections from clause 7.7 of the IEEE 802.15.4

Clause	Title & remarks/modifications	Statement
7.7	Message sequence chart illustrating MAC-PHY interaction - Figure 78: Replaced by clause 4.7.1 of this document - Figure 79: N/R - Figure 80: N/R - Figure 81: N/R - Figure 82: N/R - Figure 83: Replaced by clause 4.7.2 of this document - Figure 84 & Figure 85: Replaced by clause 4.7.3 of this document - Figure 86: N/R - Additional figure about channel estimation in clause 4.7.4 of this document	S, E

7.3.7.2 Extensions to IEEE 802.15.4 clause 7.7: Message Sequence Chart Illustrating MAC

7.3.7.2.1 PAN start message sequence chart for PAN coordinators

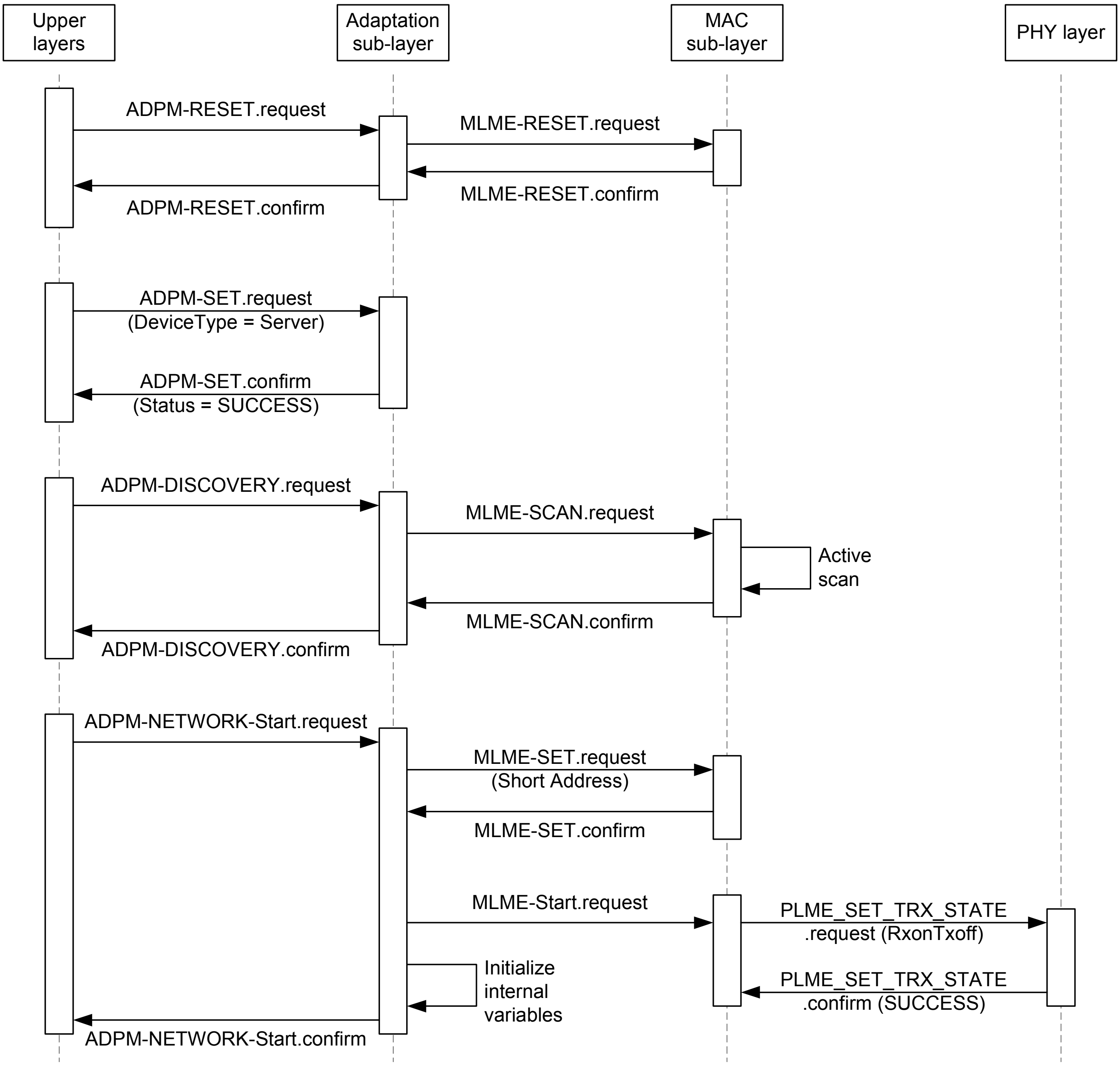
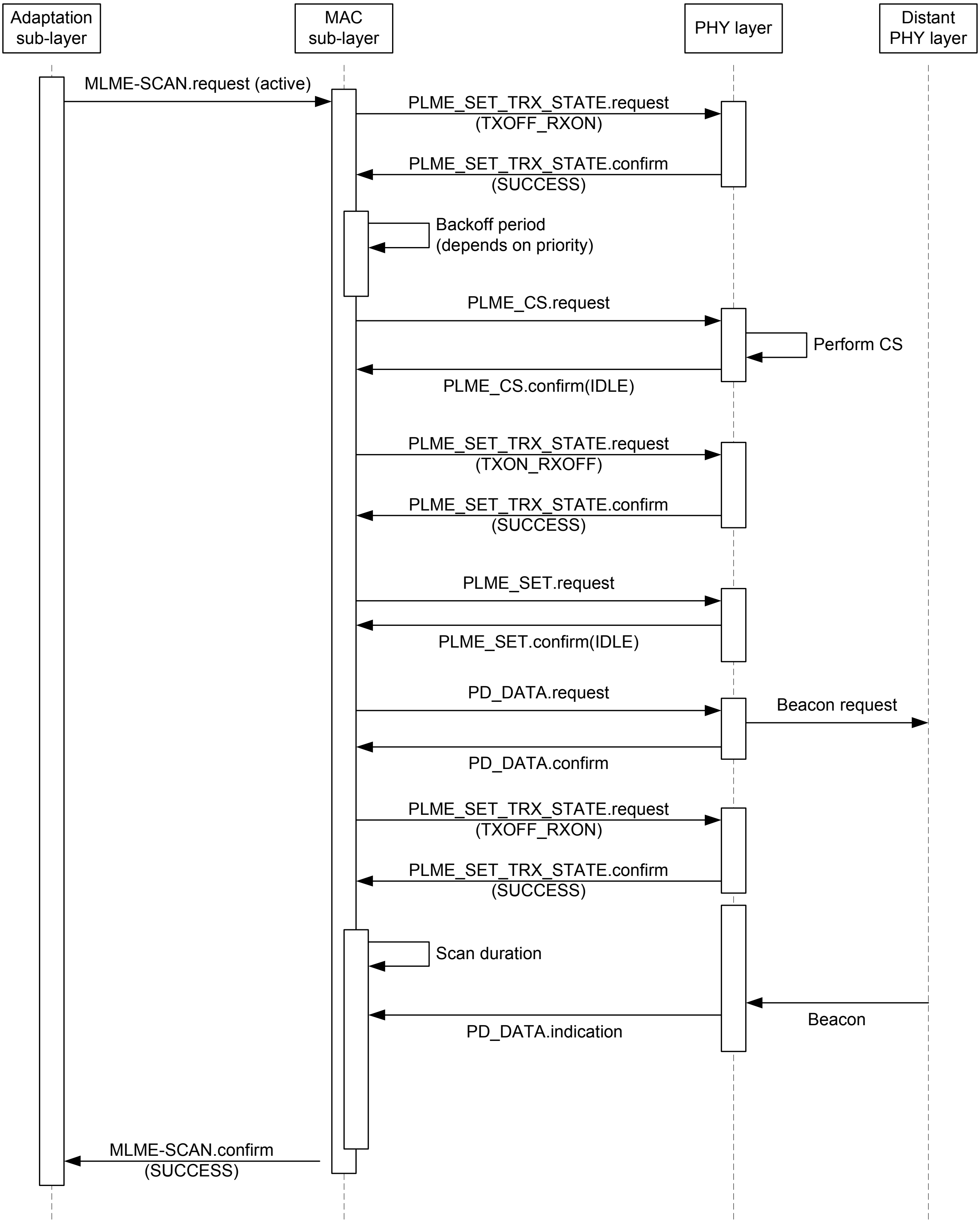


Figure 22 – PAN start message sequence chart

7.3.7.2.2 Active Scan message sequence chart



1305
1306

1307 **Figure 23 – Active scan message sequence chart**

7.3.7.2.3 Data Transmission message sequence chart

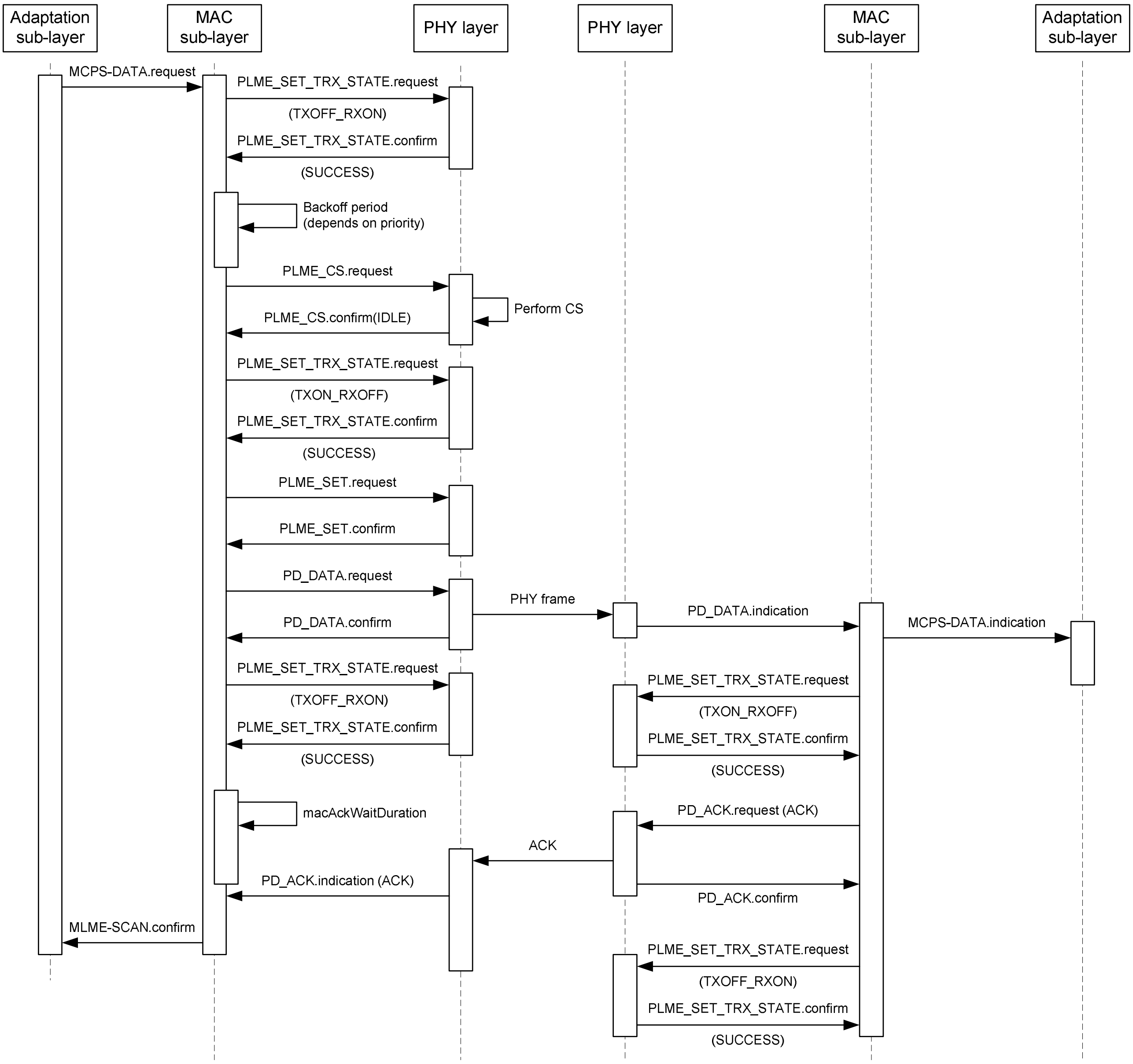


Figure 24 – Data transmission message sequence chart

7.3.7.2.4 Channel estimation message sequence chart

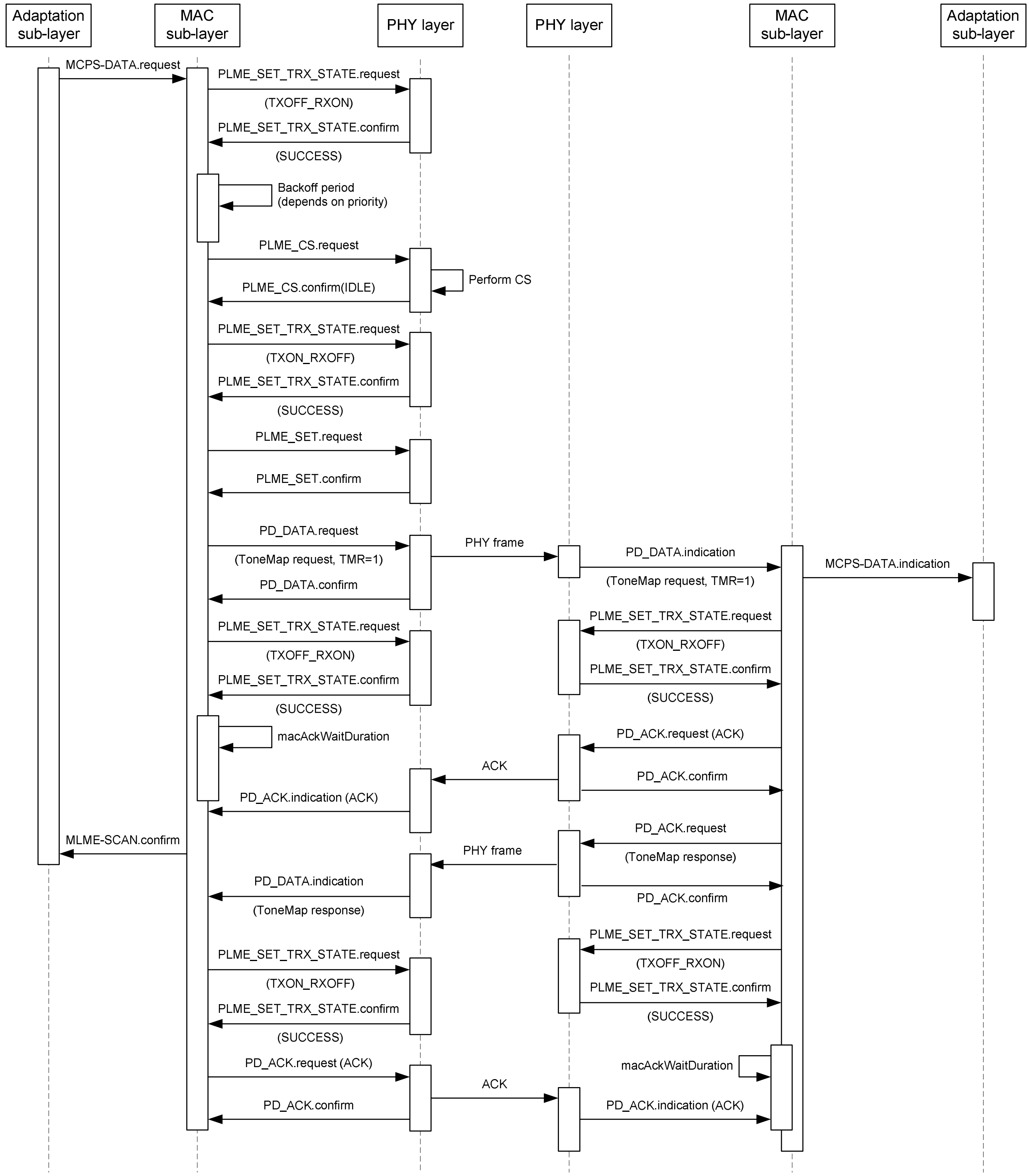


Figure 25 – Channel estimation (tone map request) message sequence chart

7.3.8 MAC annexes (based on IEEE 802.15.4 annexes)

The MAC annexes of IEEE 802.15.4-2006 apply, with the selections specified in Table 45.

Table 45 – Selections from MAC annexes of the IEEE 802.15.4

Clause	Title & remarks/modifications	Statement
Annex A	Service-specific convergence sublayer (SSCS) - IEEE 802.2 convergence sublayer is not used in the present specification	N/R
Annex B	CCM* mode of operation	N
Annex C	Test vectors for cryptographic building blocks	N
Annex D	Protocol implementation conformance statement (PICS) - The protocol implementation conformance tables are given in Annex B.	E
Annex E	Coexistence with other IEEE standards and proposed standards - This annex relates to wireless PHY standards and is not relevant for PLC technology	N/R
Annex F	IEEE 802.15.4 regulatory requirements - This annex relates to wireless PHY standards and is not relevant for PLC technology	N/R

7.4 Adaptation sublayer specification

7.4.1 Services and primitives

The Services and Primitives of the adaptation sublayer are described in Annex G.

7.4.2 Information base attributes

7.4.2.1 General

Table 46 lists the Information Base (IB) attributes of the adaptation sublayer.

Table 46 – Adaptation sublayer IB attributes

Attribute	Identifier	Type	Read Only	Range	Description	Default
adpIPv6Address	0x01	IPv6 address	Yes	Any	Defines the IPv6 address obtained from adpShortAddress	FE80::::FFFF:00FF:FE00:FFFF
adpBroadcastLogTableEntryTTL	0x02	Unsigned Integer	No	0-3 600	Defines the time while an entry in the adpBroadcastLogTable remains active in the table (in seconds).	10
adpDiscoveryAttemptsSpeed	0x06	Unsigned Integer	No	1-3 600	Allows programming the maximum wait time between two successive network discoveries (in seconds).	60
adpPANConflictWait	0x08	Unsigned Integer	No	0-3 600	Defines the time to wait between two consecutive CONFLICT frames for the same conflicting PAN ID (in seconds).	1 800
adpMaxPANConflictCount	0x09	Unsigned Integer	No	0-100	Defines the maximum number of CONFLICT frames sent by a device for the same PAN ID.	3
adpActiveScanDuration	0x0A	Unsigned Integer	No	0-60	Defines the time while an active scan must last (in seconds).	5
adpBroadcastLogTable	0x0B	Set	Yes	-	Contains the broadcast log table, see 7.4.2.2 and 7.4.4.2.2.1.	Empty
adpRoutingTable	0x0C	Set	Yes	-	Contains the routing table, see clause 5.1 in draft-daniel-6lowpan-load-adhoc-routing-03.	Empty
macNeighborTable	0x0D	Set	Yes	-	Contains the neighbour table, see 7.3.5.2.	Empty

Attribute	Identifier	Type	Read Only	Range	Description	Default
adpGroupTable	0x0E	Set	No	-	Contains the group addresses to which the device belongs.	Empty
adpToneMask	0x0F	70 bits	No	Any	Defines the Tone Mask to use during symbol formation	All bits set to 1
adpMaxHops	0x10	Unsigned Integer	No	1-8	Defines the maximum number of hops to be used by the routing algorithm.	4
adpDeviceType	0x11	Unsigned Integer	No	0-2	Defines the type of the device connected to the modem: 0: Device, 1: Server, 2: Not_Device,Not_Server	2
adpNetTraversalTime	0x12	Unsigned Integer	No	0-3 600	The Max duration between RREQ and the correspondent RREP (in seconds)	3 000
adpRrtTtl	0x13	Unsigned Integer	No	0-3 600	The time to live of a route request table entry (in seconds)	10
adpKr	0x14	Unsigned Integer	No	0-31	The Kr constant to calculate the route cost.	6
adpKm	0x15	Unsigned Integer	No	0-31	The Km constant to calculate the route cost.	5
adpKc	0x16	Unsigned Integer	No	0-31	The Kc constant to calculate the route cost.	5
adpKq	0x17	Unsigned Integer	No	0-31	The Kq constant to calculate the route cost.	5
adpKh	0x18	Unsigned Integer	No	0-31	The Kh constant to calculate the route cost.	5
adpRREQRetries	0x19	Unsigned Integer	No	Any	The number of RREQ retransmission in case of RREP reception time out.	3
adpRREQRERRWait	0x1A	Unsigned Integer	No	Any	The number of seconds to wait between two consecutive RREQ\RRER generations.	400
adpWeakLQIValue	0x1B	Unsigned Integer	No	Any	The weak Link Value defines the threshold below which a direct neighbour is not taken into account during the commissioning procedure (compared to the LQI measured).	63
adpKrt	0x1C	Unsigned Integer	No	0-31	The Krt constant to calculate the route cost.	5
adpSoftVersion	0x1D	Set	Yes	-	The soft version	-
adpSpyMode	0x1E	Unsigned Integer	No	0-1	Spy Mode activation/deactivation	0

7.4.2.2 Broadcast log table entry

Table 47 describes the broadcast log table entry:

Table 47 – Broadcast log table entry

Size → 16 bits	8 bits	13 bits
Source Address	Sequence Number	TTL

7.4.3 Data frame format, datagram transmission and addressing (based on RFC 4944)

7.4.3.1 Selections from RFC 4944

The data frame format, the theory of operation for datagram transmission using the IEEE 802.15.4 MAC sublayer, and the addressing scheme as specified in RFC 4944 together with the selections listed in Table 48.

Table 48 – Selections from RFC 4944

Clause	Title & remarks/modifications	Statement
1.	Introduction	N
1.1.	Requirements Notation	N
1.2.	Terms Used	N
2.	IEEE 802.15.4 Mode for IP - Data frames should always be acknowledged - Only non beacon-enabled network are used	S
3.	Addressing Modes - IPv6 prefixes learning via router advertisements is not supported	S
4.	Maximum Transmission Unit	N
5.	LoWPAN Adaptation Layer and Frame Format - Extension: additional Command Frame header: see 7.4.3.2.1. - When more than one LoWPAN header is used in the same packet, they MUST appear in the following order: Mesh Addressing Header Broadcast Header Fragmentation Header Command Frame Header (see 7.4.3.2.1)	E
5.1.	Dispatch Type and Header	N
5.2.	Mesh Addressing Type and Header - The value of the HopsLeft field must not exceed adpMaxHops (see 7.4.2.1).	S
5.3.	Fragmentation Type and Header	N
6.	Stateless Address Autoconfiguration - The Interface Identifier (see RFC 4291) for an IEEE 802.15.4 interface MUST be based on the EUI-64 identifier assigned to the device, the latest being itself based on a EUI-48. - Additional care must be taken when choosing a PAN identifier, so that not to interfere with I/G and U/L bits of the interface identifier. If the PAN identifiers are chosen randomly, then should be logically ANDed with 0xFCFF	S
7.	IPv6 Link Local Address	N
8.	Unicast Address Mapping	N
9.	Multicast Address Mapping	N
10.	Header Compression	N
10.1.	Encoding of IPv6 Header Fields	N
10.2.	Encoding of UDP Header Fields	N
10.3.	Non-Compressed Fields	N
10.3.1.	Non-Compressed IPv6 Fields	N
10.3.2.	Non-Compressed and Partially Compressed UDP Fields	N
11.	Frame Delivery in a Link-Layer Mesh - All devices must be FFD	S

Clause	Title & remarks/modifications	Statement
&11.1.	LoWPAN Broadcast	N
12.	IANA Considerations	N
13.	Security Considerations	N
14.	Acknowledgements	N/R
15.	References	N/R
15.1.	Normative References	N
15.2.	Informative References	I
Appendix A.	Alternatives for Delivery of Frames in a Mesh	N/R

7.4.3.2 Extensions to RFC 4944

7.4.3.2.1 Command Frame Header

In addition of the LoWPAN header specified in the RFC 4944, the present standard defines a new one: Command Frame header. This is used for the mesh routing procedure defines in 7.4.4.

As shown in Figure 26, the ADP sublayer command frames are identified using the ESC header type (see clause 5.1 of RFC 4944), followed by an 8-bit dispatch field indicating the type of ADP command. This header must always be in the last position if more than one header is present in the 6LowPAN frame.

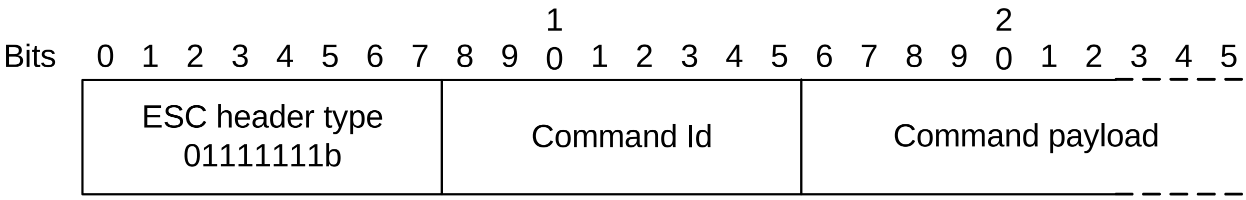


Figure 26 – Command frame header format

The ADP sublayer command frames are specified in Table 49:

Table 49 – Command frame header identifier

Command	Command Id	Comments	Specified in…
Mesh routing message	0x01	Use for mesh routing protocol	Clause 7.4.4
LoWPAN Bootstrapping Protocol message	0x02	Use for LoWPAN Bootstrap procedure	Clause 7.4.5
Contention Free Access Command	0x03	Optional	Clause 7.4.3.2.2

7.4.3.2.2 Contention Free Access command

Contention Free Access procedure is an optional feature of this standard and described in D.4.

The adaptation layer generates CFA (Contention Free Access) command if it receives an ADPD-DATA.request primitive with QualityOfService = 2 (See G.1.2). Figure 27 defines the format of the CFA command (see also Table 50).

Bits 0–7	Bits 8–15	Bits 16–23
ESC header type 01111111b	CFA command 00000011b	CFA value

Figure 27 – CFA command format

Table 50 – CFA value field description

CFA value	Description
0	Request to allow a transmission during contention free slot
1	Request to stop a transmission during contention free slot
2	Response with SUCCESS
3	Response with FAIL

The network coordinator may always use a contention free slot for transmission if other devices are not allowed to use it the same time. Other devices must ask the network coordinator for permission to use a contention free slot (CFS) for transmission by sending CFA command with request. The network coordinator may allow requested device to use a CFS for transmission by sending a confirmation response. After receiving a successful response from the network coordinator requested device can start a transmission during CFS. If the network coordinator denies a request the device should not use a CFS for transmission. The requested device must send a request to stop using CFS when it is done with contention free transmission.

Priority management can be performed using the “Normal” and “High” priority values for QOS parameter of MCPS-DATA.request primitive.

7.4.4 Mesh Routing (based on draft-daniel-6lowpan-load-adhoc-routing-03)

7.4.4.1 Selections from draft-daniel-6lowpan-load-adhoc-routing-03

The mesh routing as described in draft-daniel-6lowpan-load-adhoc-routing-03 applies, with the selections specified in Table 51.

Table 51 – Selections from draft-daniel-6lowpan-load-adhoc-routing-03

Clause	Title & remarks/modifications	Statement
1.	Introduction	N
2.	Requirements notation	N
3.	Overview - Routing is only permitted with 16-bit addresses, - LOAD uses the route cost described in Annex C as a metric of routing.	S,E
4.	Terminology	N
5.	Data Structures	N
5.1	Routing Table Entry - The destination address must be a 16-bit address, - The next hop address must be a 16-bit address, - The routing table is stored in the IB under the attribute adpRoutingTable.	S, E
5.2	Route Request Table Entry - The originator address must be a 16-bit address, - The reverse route address must be a 16-bit address.	S

Clause	Title & remarks/modifications	Statement
5.3	Message Format - For path discovery procedure, two messages have been added: Path Request (PREQ) and Path Reply (PREP). See 7.4.4.2.4.	E
5.3.1	Route Request (RREQ) - The CT field must be equal to 0x0F, to specify the use of the route cost described in Annex C, - The D bit must be set to 1, - The O bit must be set to 1, - The link layer destination and originator address must be 16-bit addresses.	S
5.3.2	Route Reply (RREP) - The CT field must be equal to 0x0F, to specify the use of the route cost described in Annex C, - The D bit must be set to 1, - The O bit must be set to 1, - The link layer destination and originator address must be 16-bit addresses.	S
5.3.3	Route Error (RERR) - The D bit must be set to 1, - The O bit must be set to 1, - The unreachable address must be 16-bit addresses.	S
6.	Operation	N
6.1	Generating Route Request	N
6.2	Processing and Forwarding Route Request	N
6.3	Generating Route Reply	N
6.4	Receiving and Forwarding Route Reply	N
6.5	Local Repair and RERR - If a link break occurs or a device fails during the delivery of data packets, the upstream node of the link break MUST repair the route locally, and execute the repairing procedure described in the present clause.	S
7.	Configuration Parameters - The values of the configuration parameters must be: NET_TRAVERSAL_TIME = 4 000 RREQ_RETRIES = 3 WEAK_LQI_VALUE = 63 - Extension: the following parameters are added by the present standard: RREQ_RERR_WAIT = 2s PATH_DISCOVERY_TIME = 1 000	S, E
8.	IANA Consideration	N
9.	Security Considerations	N/R
10.	Acknowledgments	N/R
11.	References	N
11.1	Normative Reference	N
11.2	Informative Reference	I

7.4.4.2 Extensions to draft-daniel-6lowpan-load-adhoc-routing-03

7.4.4.2.1 Unicast Packet Routing

The routing of unicast packet is performed using the following algorithm on reception of a MCPS-DATA.indication from the MAC layer:

IF (MAC destination address == address of device)

IF (6LoWPAN destination address == 6LoWPAN address of device)

- Generate an ADPD-DATA.indication primitive to indicate the arrival of a frame to the upper layer, with the following characteristics (see G.1.4):
  - DstAddrMode = 0x02
  - DstAddr = 6LoWPAN destination address
  - SrcAddr = The originator address in the 6LoWPAN mesh header
  - NsduLength = length of the payload
  - Nsdu = the payload
  - LinkQualityIndicator = msduLinkQuality (see G.3.2)
  - SecurityEnabled = (SecurityLevel != 0)

ELSE IF (6LoWPAN destination address is in the neighbour table)

- Forward the packet to the destination address, by invoking an MCPS-DATA.request primitive with the destination address set to the final destination address

ELSE IF (6LoWPAN destination address is in the routing table and next hop in the neighbour table)

- Forward the packet to the next hop found in the routing table, by invoking an MCPS-DATA.request primitive, and using the communication parameters to that device contained in the neighbour table.

ELSE IF (6LoWPAN Destination Address not in routing table)

- Perform a link repair as described in clause 6.5 of draft-daniel-6lowpan-load-adhoc-routing-03
- Queue the packet for a sending retry

ELSE

- Drop the frame

ELSE IF (MAC Destination address == 0xFFFF)

- This is a broadcast frame: execute algorithm described in clause 7.4.4.2.2 of the present document

ELSE

- Drop the frame

7.4.4.2.2 Multicast / Broadcast

7.4.4.2.2.1 Packet Routing

The packet routing mechanism is based on clause 11.1 of RFC 4944. This clause details more precisely the routing of broadcast and multicast packets.

As described in clause 11.1 of RFC 4944, each broadcast packet has a BC0 header containing a sequence number. Each time a node sends a broadcast packet, it must increment this sequence number.

Each node must have a broadcast log table. This table is used for routing broadcast packets, and each entry contains the parameters described in Table 52:

Table 52 – Broadcast log table

Field Name	Size	Description
SrcAddr	2 bytes	The 16-bit source address of a broadcast packet. This is the address of the broadcast initiator.

Field Name	Size	Description
SeqNumber	Integer, 1 byte	The sequence number contained in the BC0 header
TimeToLive	Integer	The remaining time to live of this entry in the Broadcast Log Table, in milliseconds.

Each time a device receives a broadcast address with a HopsLft field of mesh header (see clause 5.2 of RFC 4944) strictly greater than 0, it must check if an entry already exists in the broadcast log table having the same SrcAddr and SeqNumber. If an entry exists, the received frame is silently discarded. Else, a new entry is added in the table, and the TimeToLive field is initialized with the value adpBroadcastLogTableEntryTTL (see 7.4.2). When this value reaches 0, the entry is removed from the broadcast log table.

When a device receives a broadcast frame, so that it has to create an entry in the broadcast log table, it must decrement its HopsLft field and trigger the emission of the received broadcast frame using CSMA/CA. The frame will then be sent as if it was a standard unicast frame using CSMA/CA.

This can be summarized by the following algorithm, executed upon reception of a frame whose destination address is 0xFFFF:

IF (HopsLft == 0)

- Discard frame and exit

ELSE IF ((SrcAddr, SeqNumber) exists in broadcast log table)

- Discard frame

ELSE

- Create one entry (SrcAddr, SeqNumber, adpBroadcastLogTableEntryTTL) in broadcast log table, with the corresponding frame characteristics.

IF (final destination address = broadcast address) or (final destination address is found in adpGroupTable)

- Generate an ADPD-DATA.indication primitive to upper layer with the following characteristics:
    - DstAddrMode = 0x01
    - DstAddr = Destination address in the 6LoWPAN mesh header (multicast or broadcast address)
    - SrcAddr = The originator address in the 6LoWPAN mesh header
    - NsduLength = length of the data
    - Nsdu = the data
    - LinkQualityIndicator = msduLinkQuality (see G.3.2)
    - SecurityEnabled = (SecurityLevel != 0)
- Trigger the sending of the frame using CSMA/CA

NOTE In case of a multicast address, the broadcast address 0xFFFF is used at the MAC level as mentioned in clause 3 of RFC 4944. Multicast frames are routed using the same algorithm as broadcast frames.

The broadcast log table is available in the Information Base with the attribute adpBroadcastLogTable (see 7.4.2).

7.4.4.2.2.2 Groups

Each device can belong to one or more group of devices. The IB attribute adpGroupTable (see 7.4.2) stores a list of 16-bit group addresses.

When the device receives a MAC broadcast message, and if the final destination address in the 6LoWPAN mesh header is equal to one of the 16-bit group addresses in adpGroupTable, then an ADPD-DATA.indication primitive is generated to the upper layer (as described in 7.4.4.2.2.1).

Groups can be added or removed from the adpGroupTable using ADPM-SET.request primitive. The size of this table is implementation specific, and must have at least one entry. The way groups are managed by upper layers is beyond the scope of this document.

7.4.4.2.3 Route Discovery

7.4.4.2.3.1 Manual Route Discovery

A manual route discovery can be triggered by the upper layer, for maintenance or performance purposes. This is done through the invocation of the ADPM-ROUTE-DISCOVERY.request primitive. The adaptation sublayer then generates a RREQ frame and executes the algorithms as described in 7.4.4.1.

After the algorithm completes, the adaptation sublayer generates an ADPM-ROUTE-DISCOVERY.confirm primitive with the corresponding status code, and eventually modify its routing table.

Only one route discovery procedure can be processed in the same time. All other ADPM-ROUTE-DISCOVERY.request will be ignored.

All devices are required to handle RREQ, RREP and RERR frames as described in 7.4.4.1 and must modify their routing tables accordingly.

7.4.4.2.3.2 Automatic Route Discovery

If an ADPD.DATA.request primitive is invoked with its DiscoverRoute parameter set to TRUE, and if no entry is available in the routing table for the device designated by DstAddr, then the adaptation layer generates a RREQ and executes the algorithms described in 7.4.4.1 in order to find a route to the destination. If the route discovery succeeds, then the data frame is send to the destination according to the newly discovered route. If the route discovery fails, then the adaptation layer must generate an ADPD-DATA.confirm primitive with the status code ROUTE_ERROR.

If an ADPD.DATA.request primitive is invoked with its DiscoverRoute parameter set to FALSE, and if no entry is available in the routing table for the device designated by DstAddr, then the adaptation layer must generate an ADPD-DATA.confirm primitive with the status code ROUTE_ERROR.

Route repairing procedures are described in 7.4.4.1.

7.4.4.2.3.3 RREQ RERR Generation Frequency Limit

A node must wait RREQ_RERR_WAIT second between two successive RREQ/RERR generations to limit the number of broadcast packet in the network. The definition of the RREQ_RERR_WAIT parameter is given in 7.4.4.1.

7.4.4.2.4 Path Discovery

7.4.4.2.4.1 Operation

A path discovery can be triggered by the upper layers, for maintenance or performance purposes. This is done through the invocation of the ADPM-PATH-DISCOVERY.request primitive. The adaptation sublayer then generates a PREQ frame and executes the algorithms described in following sub-clauses.

After the algorithm completes (with the reception of a PREP frame), the adaptation sublayer generates an ADPM-PATH-DISCOVERY.confirm primitive to the upper layer.

Only one path discovery procedure can be processed in the same time. All other ADPM-PATH-DISCOVERY.request from the upper layer will be ignored.

– Generating a Path Request (PREQ)

During the path discovery period, an originator, a node that requests a path discovery, generates a Path Request (PREQ) message (see 7.4.4.2.4.2).

Once transmitted, the node waits for a Path Reply (PREP), else, and after PATH_DISCOVERY_TIME milliseconds, the node generate an ADPM-PATH-DISCOVERY.confirm to the upper layer with an NsduId field containing a Path Reply (PREP) with HOPS fields set to 0.

– Processing and forwarding a Path Request (PREQ)

Upon receiving a Path Request (PREQ), an intermediate node tries to find entry of the same destination address in the routing table. If the entry is found, the node just forwards the PREQ to the next hop toward the destination. Else, the node just discards the PREQ.

– Generating a Path Reply (PREP)

A final node, on receiving a Path Request (PREQ), generates a Path Reply (PREP) with the following information:

- Flag R set to 0,
- Hops field set to 1,
- Hops1 address set to its own address.

If an intermediate node can't find a route to the destination of the PREQ, it generates a Path Reply (PREP) with R flag set to 1, the HOP to 1 and the Hops1 address to its own address then sends it to the source of the PREQ.

– Processing and forwarding a Path Reply (PREP)

Upon receiving a Path Reply (PREP), an intermediate node tries to find entry of the same destination address in its own routing table. If the entry is found, the node just forwards the PREP to the next hop toward the destination with the following field updates:

- Set the Hop_N address field in the PREP to its own address, with N = HOPS+1,
- Update the RC field with its own route cost and,
- Increment the HOPS field by one.

If there's no route to the destination, the node just discard the Path Reply message received.

7.4.4.2.4.2 Path Request Frame

The path request frame format and the detail of its related fields are described in Figure 28 and Table 53 respectively.

Bits	0–7	8–23	24–31
	Type	Destination Address	Originator...
	… Address		

Figure 28 – Path Request (PREQ) message format

Table 53 – Path Request (PREQ) fields' definition

Field	Size, bits	Value	Definition
Type	8	4	Path Request (PREQ) message identifier
Destination Address	16	-	The 16 bit short link layer address of the destination for which a route is supplied
Originator Address	16	-	The 16 bit short link layer address of node which originated the packet.

7.4.4.2.4.3 Path Reply Frame

The path reply frame format and the detail of its related fields are described in Figure 29 and Table 54 respectively.

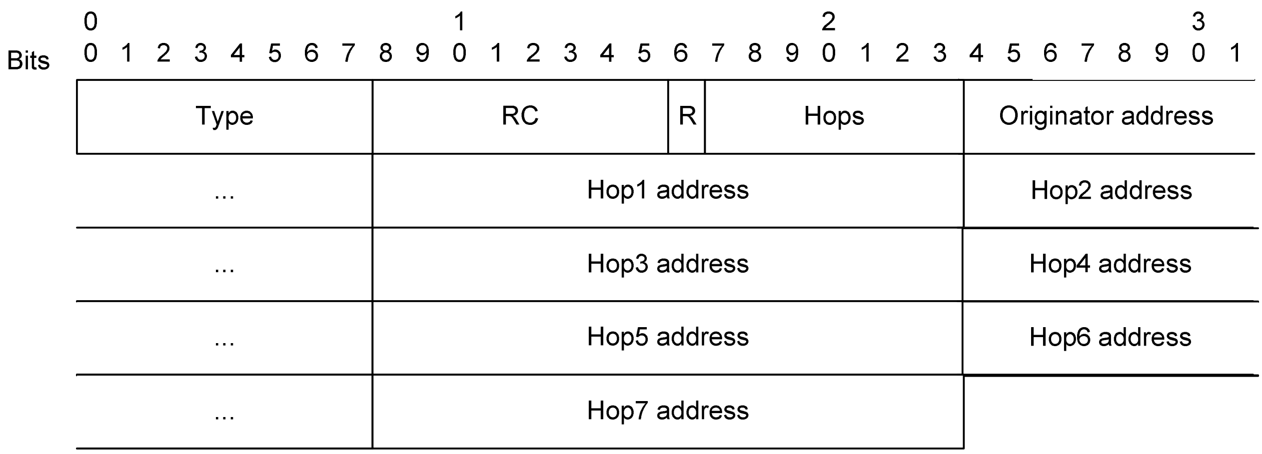


Figure 29 – Path Reply (PREP) message format

Table 54 – Path Reply (PREP) fields' definition

Field	Size (bits)	Value	Definition
Type	8	4	Path Request (PREQ) message identifier
RC	8		Route Cost - The accumulated link cost of the reverse route from the originator to the sender of the message.
R	1	0/1	Path discovery result: 1 Success of path discovery 0 Failure of path discovery
Hops	7	-	Number of hops of the route
Originator Address	16	-	The 16 bit short link layer address of node which originated the packet.
Hop_N	16	-	The 16 bit short link layer address of nodes constituting the path.

7.4.5 Commissioning of New Devices (based on draft-6lowpan-commissioning-02)

7.4.5.1 Selections from draft-6lowpan-commissioning-02

The commissioning of new devices on an existing network as described in draft-6lowpan-commissioning-02 applies, with the selections specified in Table 55.

Table 55 – Selections from draft-6lowpan-commissioning-02

Clause	Title & remarks/modifications	Statement
1.	Introduction	N
2.	Terminology	N
2.1.	Requirements notation	N
3.	Bootstrapping - Obtaining a 16-bit short address and security credentials are mandatory parts of the commissioning process.	S
3.1.	Resetting the device	N
3.2.	Scanning through channels - For getting the information of other devices within POS, the device MUST perform an active scan.	S
3.3.	LoWPAN Bootstrapping Mechanism -'LBA discovery phase' is described in 7.4.5.2.2	E
3.3.1.	LoWPAN Bootstrapping Protocol message format	N
3.3.1.1	LBP message - Some enhancements and clarifications to LBP message format are given in 7.4.5.2.1.	E
3.3.2.	LoWPAN Bootstrapping Information Base - PAN_type must always be Secured - Address_of_LBS must be equal to the default address of the PAN coordinator that is 0x0000. - Short_Addr_Distribution_Mechanism must be 0 for centralized address management.	S
3.3.3.	LBA discovering phase - Some enhancements and clarifications to 6LoWPAN bootstrapping procedure are given in 7.4.5.2.2. - The LBD must perform an active scan instead of broadcasting a LBA solicitation message.	E
3.3.4.	LoWPAN Bootstrapping Protocol (LBP)	S
3.3.5.	Bootstrapping in open 6LoWPAN	N/R
3.3.6.	LBP in secured 6LoWPAN - The LBP messages from the LBD to the LBA are sent by invocation of the ADPD-DATA.request primitive with the following attributes: - DstAddrMode = 0x02 - DstAddr = The MAC address of the LBA passed as an argument to the ADPM-NETWORK-JOIN.request primitive - NsduLength = the length of the LBP message - Nsdu = the LBP message itself - NsduHandle = random number - MaxHops = 0 - DiscoverRoute = FALSE - QualityOfService = 0 - SecurityEnabled = FALSE NOTE The LBA is already present in the neighbour table because an active scan must have been performed prior to invoking the ADPM-NETWORK-JOIN.request primitive. Thus the routing algorithm will operates correctly as described in 7.4.4 of the present document. - The LBP messages from the LBA relayed to the LBS are sent by invocation of the ADPD-DATA.request primitive with the following attributes: - DstAddrMode = 0x02	S

Clause	Title & remarks/modifications	Statement
	- DstAddr = The MAC address of the LBS - NsduLength = the length of the LBP message - Nsdu = the LBP message itself - NsduHandle = random number - MaxHops = adpMaxHops (see 7.4.2.1) - DiscoverRoute = TRUE - QualityOfService = 0 - SecurityEnabled = TRUE - The LBP messages from the LBS to the LBD relayed to the LBA are sent by invocation of the ADPD-DATA.request primitive with the following attributes: - SrcAddrMode = 0x02 - DstAddrMode = 0x03 - DstPANId = the PAN ID - DstAddr = 64-bit address of the LBD - NsduLength = the length of payload - Nsdu = the payload - msduHandle = a random number - TxOptions = 100b - SecurityLevel = 0 - KeyIdMode = ignored - KeySource = ignored - KeyIndex = ignored	
3.3.7.	Role of Entities in LBP - If a LBD does not find any LBA during the LBA discovery phase, it must still perform LBA discoveries as long as it is not commissioned. Note that LBA discovery is done using active scans rather that broadcasting LBA solicitation messages. - Only secured networks are used	S
3.4.	Assigning the short address Short addresses are assigned in a centralized fashion by the LBS	S
3.5.	Obtaining IPv6 address - The devices do not need to obtain an IPv6 address prefix, and the procedures described in this clause as well as in RFC 4862 must be ignored. Only the IPv6 Link Local Address generated as stated in clause 7 of RFC 4944 is used for communication.	M
3.6.	Configuration Parameters - The values of the configuration parameters must be: CHANNEL_LIST = 0xFFFF800 (not used) SCAN_DURATION = adpActiveScanDuration (see 7.4.2.1) SUPERFRAME_ORDER = 15 BEACON_ORDER = 15 START_RETRY_TIME = 0 (not used) JOIN_RETRY_TIME = 0 (not used) ASSOCIATION_RETRY_TIME = 0 (not used)	M
4.	IANA Consideration	N/R
5.	Security Considerations	N
6.	Contributors	N/R
7.	Acknowledgments	N/R

Clause	Title & remarks/modifications	Statement
8.	References	N
8.1.	Normative References	N
8.2.	Informative References	I

7.4.5.2 Extensions to draft-6lowpan-commissioning-02

7.4.5.2.1 LoWPAN Bootstrapping Protocol (LBP) message format

7.4.5.2.1.1 General

LBP message format and the detail of its related fields are described in Figure 30 and Table 56 respectively.

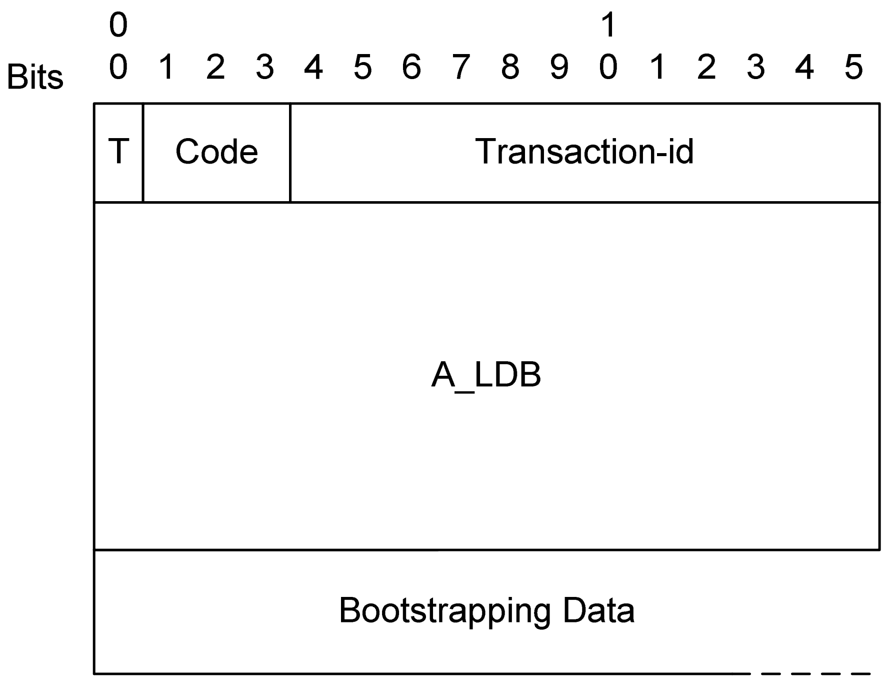


Figure 30 – LBP message format

where

T identifies the type of message (1-bit)

0 Message from LDB

1 Message to LDB

Code identifies the message code (3-bit) defined in Table 56.

Transaction-id aids in matching Responses with Request (12-bit)

A_LDB The A_LDB field is 8 octets and indicates the EUI-64 address of the bootstrapping device (LBD).

Bootstrapping Data The Bootstrapping Data field is of variable length and contains additional information elements. Two types are defined:

– Embedded EAP messages (see 7.4.5.2.1.2),
– Configuration parameters (see 7.4.5.2.1.3).

Table 56 – T & Code fields in LBP message

T	Code	LDB message	Description
0	001	JOINING	The LDB requests joining a PAN and provides necessary authentication material
1	001	ACCEPTED	Authentication succeeded with delivery of Device Specific Information (DSI) to the LDB
1	010	CHALLANGE	Authentication in progress. PAN Specific Information (PSI) may be delivered to the LDB
1	011	DECLINE	Authentication failed
0/1	100	KICK	KICK frame is used by a PAN coordinator to force a device to loose its MAC address, or by any device to inform the coordinator that it left the PAN. On reception of this frame, a device must set its short address to the default value of 0xFFFF, disconnect itself from the network, and perform a reset of the MAC and adaptation layers. See 7.4.5.2.2.7for details about kicking procedure.
0	101	CONFLICT	CONFLICT frame is used by a device to inform the PAN coordinator that it has detected another PAN operating in the same POS. See 7.5.2 for details about PAN ID conflict handling.

7.4.5.2.1.2 Embedded EAP messages

LBP messages embed Extended Authentication messages (EAP) as defined by RFC 3748. Figure 31 describes minor modification to fit the generic LBP information element format.

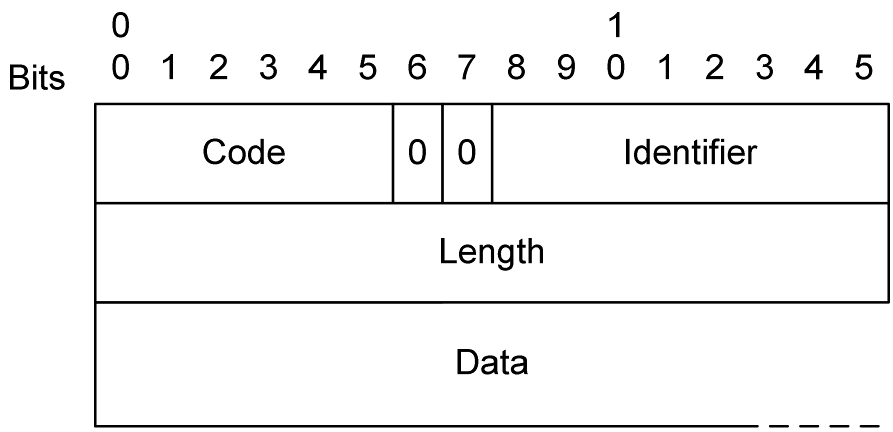


Figure 31 – Embedded EAP message format (generic)

where

Code identifies the Type of EAP packet (6-bit). EAP Codes are assigned as follows:

1 Request (sent to the peer = LDB)
2 Response (sent by the peer)
3 Success (sent to the peer)
4 Failure (sent to the peer)

The Code field is slightly different from a regular EAP Code field as specified in RFC 3748. The conversion appears straightforward in both directions. The proper conversion must apply when the EAP message is propagated over another protocol (i.e. RADIUS) and in case of integrity protection covering the EAP header.

Identifier and aids in matching Responses with Requests (8-bit).

Length The Length field is two octets and indicates the length, in octets, of the EAP packet including the Code, Identifier, Length, and Data fields. A message with the Length field set to a value larger than the number of received octets must be silently discarded.

Data The Data field is zero or more octets. The format of the Data field is determined by the Code field. Refer to RFC 3748 for more details on:

- Specific format for Request / Response messages and the introduction of the Type field (Identity, Nak, etc.),
- Specific format for Success / Failure messages with an empty Data field.

7.4.5.2.1.3 Configuration parameters

Configuration parameter format and the detail of its related fields are described in Figure 32:

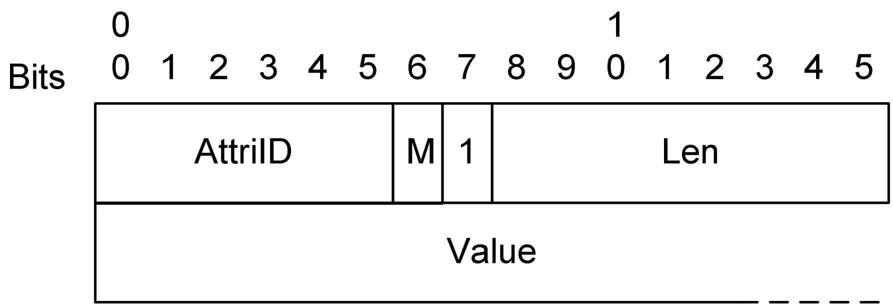


Figure 32 – Configuration parameter format

where

Attr-ID represents the ID of the Attribute in LoWPAN Information Base (LIB) (6-bit)

M identifies the type of the Attribute (1-bit):

0 Device Specific Information (DSI)

1 PAN Specific Information (PSI)

Len indicates the length, in octets, of the Value field (8-bit)

Value is zero or more octets and contains the value of the Attribute. Its format is defined by Attr-ID.

7.4.5.2.2 6LoWPAN bootstrapping procedures

7.4.5.2.2.1 Overview

This clause proposes some enhancements and clarifications to 6LoWPAN bootstrapping procedure. This procedure is executed when the ADPM-NETWORK-JOIN.request primitive is invoked by the upper layer.

Figure 33 provides an overview of the messages exchanged between devices during the Bootstrapping procedure.

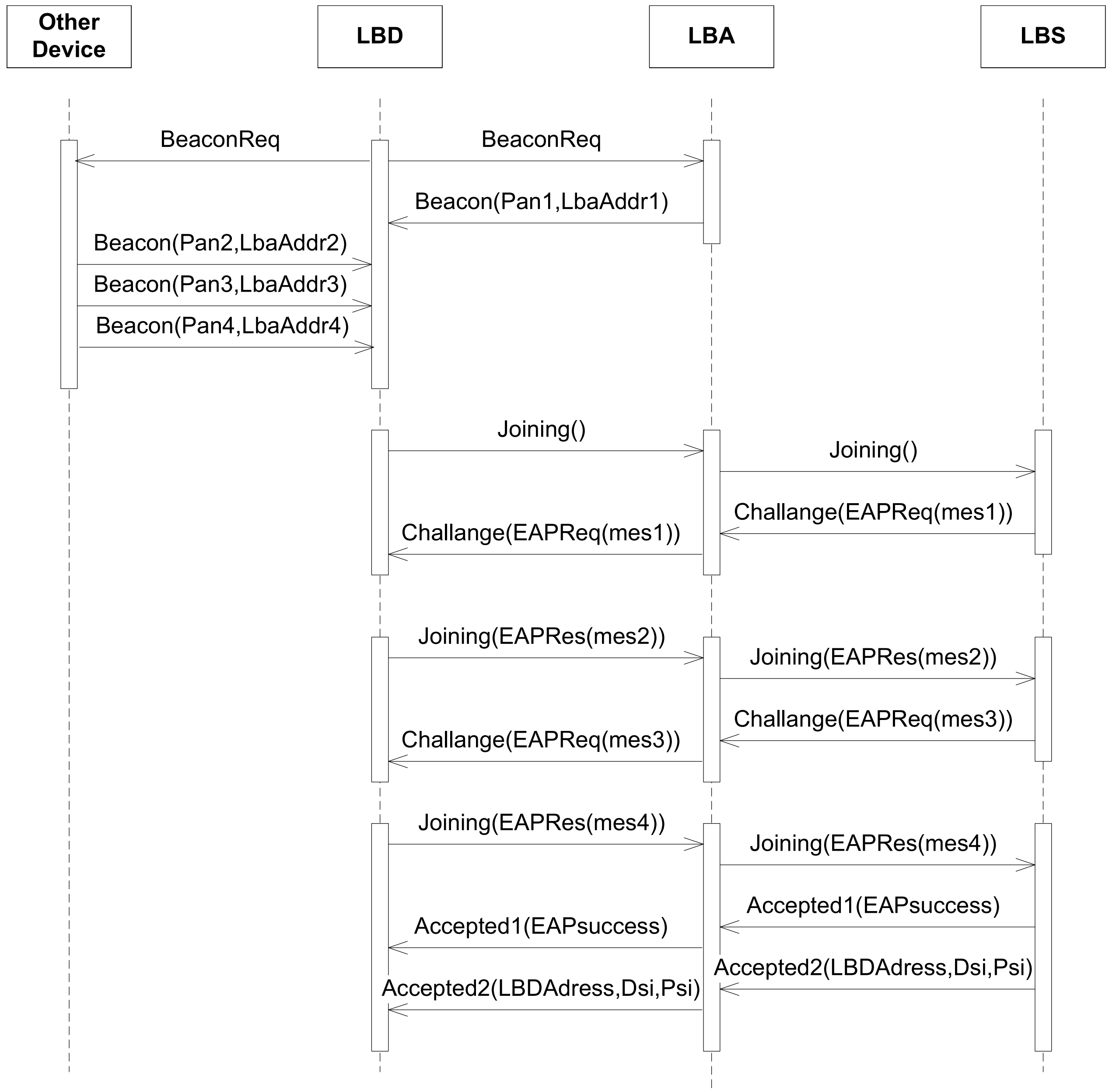


Figure 33 – Bootstrapping protocol messages sequence chart

Figure 34 summarizes the forwarded messages involved during a nominal association procedure on a PAN between different protocol layers of the devices, when a single LBP protocol message needs to be exchanged between the LBD and the LBS.

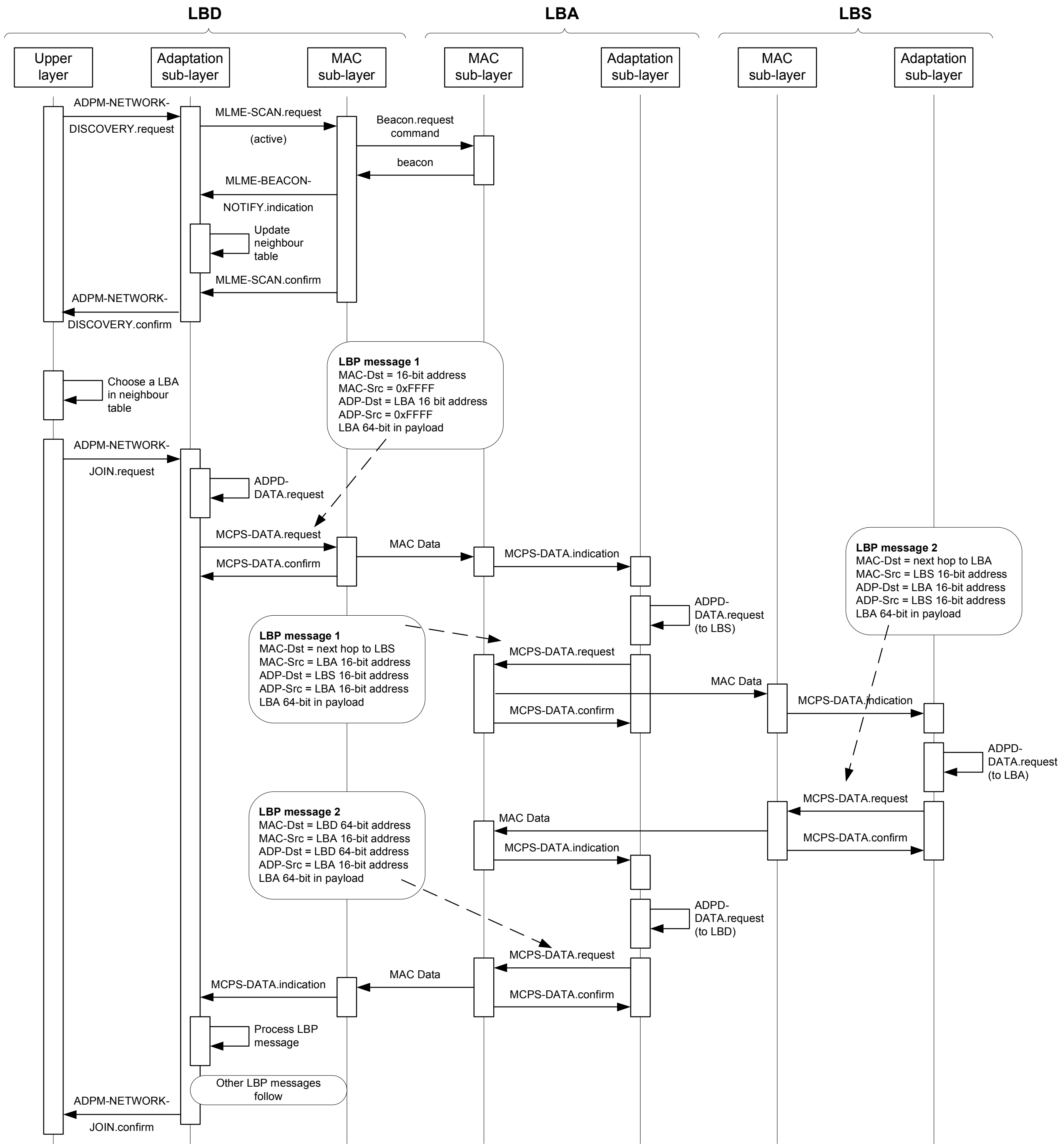


Figure 34 – Bootstrapping protocol messages forwarding

7.4.5.2.2.2 Discovering phase

At the beginning of the Bootstrapping procedure, an End Device (aka LoWPAN Bootstrapping Device or LBD) must launch an "active channel scan" (cf. IEEE 802.15.4 clause 7.5.2.1.2).

The higher layer can start an active scan by invoking the ADPM-DISCOVERY.request primitive, and specifying the duration of the scan. The adaptation layer then invokes the MLME-SCAN.request primitive of the MAC layer with the following parameters:

- ScanType = 0x01;
- ScanChannels = all bits to 0 (not used);
- ScanDuration = Duration;
- ChannelPage = 0 (not used);
- SecurityLevel = 0;
- KeyIdMode = Ignored;
- KeySource = Ignored;
- KeyIndex = Ignored.

The LBD sends a 1-hop broadcast Beacon.request frame and any Full Feature Device in the Neighbourhood should reply by sending a Beacon frame with its PAN identifier, short address and capabilities.

Upon completion, the MAC layer issues a MLME-SCAN.confirm primitive, with the list of existing PAN in the PANDescriptorList parameter. In response, the adaptation layer generates an ADPM-DISCOVERY.confirm primitive which contains the PANDescriptorList parameter provided by the MAC layer.

At the end of the scan, the LBD selects one of the Beacon senders. It may be either the PAN coordinator that play the role of LoWPAN Bootstrapping Server (LBS) or another FFD. In the latter case, the FFD (aka LoWPAN Bootstrapping Agent or LBA) is in charge of relaying the LoWPAN Bootstrapping Protocol (LBP) frames between the LBA and the LBS.

The choice is based on the following criteria:

a) Association permit, rejected if negative;
b) Link Quality, rejected below a threshold;
c) PAN identifier, according to a round robin algorithm;
d) PAN coordinator, preferred if positive;
e) Short address, according to a round robin algorithm.

A device must not perform more than adpMaxDiscoveryPerHour network discovery procedures per hour.

7.4.5.2.2.3 Access control phase

Once the discovery phase is finished, the LBD send an LBP JOINING frame to the LBA. This frame includes a field that carries the EUI-64 address of the joining LBD.

This frame, as any other frame during the initial part of the Bootstrapping process, is transmitted between the LBD and the LBA without any additional security at the MAC layer.

When received by the LBA, this frame is relayed by the LBA to the LBS. The LBA is supposed fully bootstrapped with the full capability to directly transmit any message to the LBS in a secure way.

The LBP protocol has been designed to fit two different authentication architectures:

- The authentication function is directly supported by the LBS, and in this case all the authentication material (access lists, credentials, etc.) must be loaded in the LBS or,
- The authentication function is supported by a remote (and usually centralized) AAA server, and in this case, LBS is only in charge of forwarding the EAP messages to the AAA server over a standard AAA protocol (i.e. RADIUS, RFC 2865).

The following procedure description is only based on the first architecture but extension to the second one appears straightforward.

So, when received by the LBS, the EUI-64 address should be compared with an Access Control list (white list or black list) with the following possibilities:

- This address does not fit the Access Control list and the LBS send back an LBP DECLINE message, embedding an EAP Failure message or,
- This address fit the Access Control List (or the Access Control is not implemented) and the LBS send back an LBP CHALLENGE message, embedding an EAP Request message. This latter message also carries the first authentication message. In the present version of this standard, the EAP identity phase is skipped as proposed by RFC 3748 to directly move to the authentication phase by sending the first message of the selected EAP method.
- The EAP identity phase could be reintroduced later when the need of roaming features arise.

In both cases, these messages are relayed by the LBA to the LDB.

7.4.5.2.2.4 Authentication and key distribution phase

The Authentication phase is fully dependant of the EAP method in place. The EAP protocol is very flexible and support various EAP methods (EAP-MD5, EAP-AKA, EAP-TLS, etc.). Each method is characterized by its credentials (shared secret, certificate, SIM cards, etc.) and by its signature and encryption algorithms.

Methods are ordinary based on two round-trip exchanges:

- The first one for mutual authentication and initial exchange of ciphering material; and,
- The second one for mutual control of session keys derivation.

At the end, the LBD should be equipped with two sets of session keys:

- Unicast session keys for the end-to-end security of EAP messages. These keys are timely refreshed; and
- Group session keys for a basic PAN security. These keys are shared by all the authenticated nodes in the PAN. Every MAC data frame, except those involved in the initial phases of the bootstrapping procedure, is securely transmitted with encryption and decryption at every hop. These Group keys are timely refreshed and when a node is detached from the PAN.

Other keys may be derived for additional security services provided at the Application level.

Refer to 8.5 for further details on the proposed EAP method.

7.4.5.2.2.5 Authorization and initial configuration phase

Then, two possibilities:

- The Authentication and Key Distribution process does reach completion and the LBS sends back an LBP DECLINE message, embedding an EAP Failure message and relayed by the LBA to the LBD; or,
- This process reaches completion and the LBS selects 16-bit short Address, globally defined and fully routable in the PAN. The LBS sends back an LBP ACCEPTED message, embedding an EAP Success message. At receipt of this message, the LBD activate the GMK key. A second LBP ACCEPTED message is sent by LBS embedding the global 16-bit short address and the various parameters (Device Specific and PAN Specific). These messages are relayed by the LBA to the LDB. At this stage, the LDB

owns the necessary session's keys and, the messages are securely transferred end-to-end.

At reception of the LBP message, the LBD must set-up an optimized route to the LBS with the help of the LOAD protocol (see 7.4.4).

7.4.5.2.2.6 Joining a PAN for any node except coordinator

The network joining procedure must only be performed by a device which is not a PAN coordinator, and which does not have a short address. It is triggered by invocation of the ADPM-NETWORK-JOIN.request primitive. The algorithm to execute is:

- Short_Address = 0xFFFF (= no short address)
- Current_Neighbor_index = 0
- Connected = FALSE
- While (Short_Address == 0xFFFF)
    - Wait for a random number of seconds. The number of seconds the device must wait is

      adpNumDiscoveryAttempts x Rnd

      where,

      Rnd — is a random integer value between 1 and adpDiscoveryAttemptsSpeed,

      – adpNumDiscoveryAttempts — is the number of times the ADPM-NETWORK-DISCOVERY.request primitive was called in this procedure.

      – adpNumDiscoveryAttempts must be reset to 0 when ADPM-NETWORK-DISCOVERY.request succeeds, on device start-up, and after a reset of the adaptation layer.

      – The value of adpNumDiscoveryAttempts must not be incremented anymore if it reaches 15,
    - Perform the "Discovering Phase" through invocation of an ADPM-NETWORK-DISCOVERY.request primitive.

      NOTE The Neighbour Table must be updated for each received beacon
    - If existing PANs were found
        - Select a PAN and LBA in the neighbour table (criteria definition is implementation specific),
        - While ((Connected == FALSE) && (Current_Neighbor_index < Size_of_Neighbor_Table)
            - Perform the access control (see 7.4.5.2.2.3), the authentication and key distribution (see 7.4.5.2.2.4), then the authorization and initial configuration (see 7.4.5.2.2.5) phases through invocation of the ADPM-NETWORK-JOIN.request primitive using the PAN identifier and MAC address above:
                - PANId = PANId chosen above
                - LBAAddress = Address at index Current_Neighbor_index
            - If ADPM-NETWORK-JOIN.confirm == SUCCESS then,
                - Short_Address = result of association process
                - Connected = TRUE
            - Else
                - Current_Neighbor_index ++

- While (Connected == TRUE)
  – Wait for a disconnection

7.4.5.2.2.7 Leaving a PAN – Removal of a device by the PAN coordinator

The PAN coordinator may instruct a device to remove itself from the network invoking the ADPM-LBP.request primitive, using a KICK frame. This frame is a standard LBP message frame with its Code field set to 100b. The bootstrapping data in that message should be empty.

When a device receives this message, it must check if the A_LBD field of the LBP message is its own address. If not, the message is silently discarded. Else the device must perform the following steps:

– Acknowledge the frame if necessary;

– Set its 16-bit short address to 0xFFFF;

– Generate a ADPM-NETWORK-LEAVE.indication containing the 64-bit address of the device;

– Invoke a MLME-RESET.request primitive with the SetDefaultPIB parameter set to TRUE;

– Invoke its ADPM-RESET.request primitive to reset itself.

Figure 35 describes the messages exchanged during removal of a device from the PAN by the coordinator.

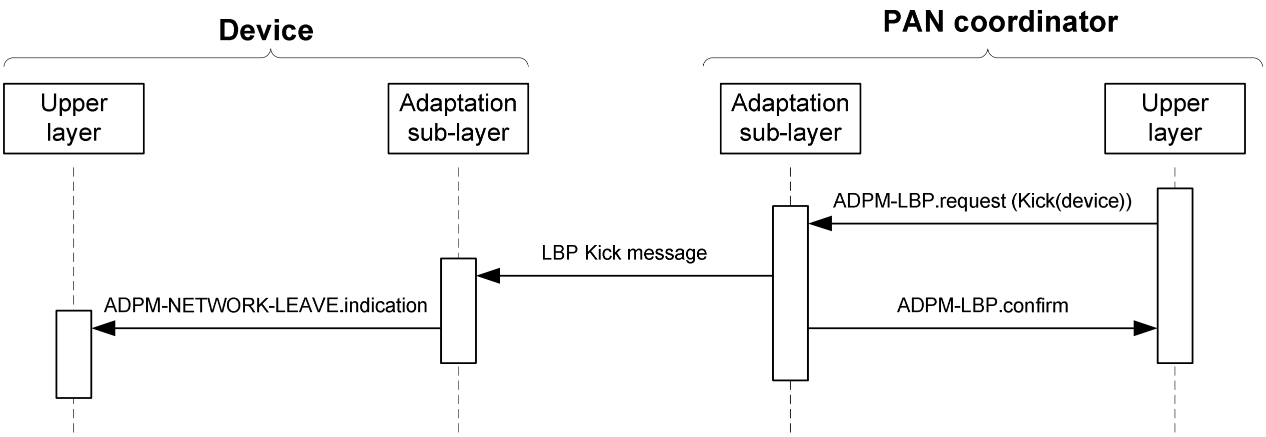


Figure 35 – Message sequence chart during removal of a device by the coordinator

Upon completion of this procedure, the device must restart the joining network procedure described in 7.4.5.2.2.

7.4.5.2.2.8 Leaving a PAN – Removal of a device by itself

A device may also call the ADPM-NETWORK-LEAVE.request primitive to remove itself from the network, and notify the PAN coordinator about this removal.

– On invocation of the ADPM-NETWORK-LEAVE.request primitive by a device which is not the PAN coordinator, and with an ExtendedAddress parameter not NULL, the adaptation sublayer must issue an ADPM-NETWORK-LEAVE.confirm primitive with the status INVALID_REQUEST;

– On invocation of the ADPM-NETWORK-LEAVE.request primitive by a device which is the PAN coordinator, and with an ExtendedAddress parameter set to NULL, the adaptation sublayer must issue an ADPM-NETWORK-LEAVE.confirm primitive with the status INVALID_REQUEST.

– On invocation of the ADPM-NETWORK-LEAVE.request primitive by a device which is not the PAN coordinator, and with a ExtendedAddress parameter set to NULL, the adaptation sublayer must:

  – Send a KICK frame to the PAN coordinator using a ADPD-DATA.request primitive (and setting the T field in the LBP message to 1, to indicate a message from LBD);
  – Set its 16-bit short address to 0xFFFF;
  – Generate a ADPM-NETWORK-LEAVE.indication containing the 64-bit address of the device;
  – Invoke a MLME-RESET.request primitive with the SetDefaultPIB parameter set to TRUE;
  – Invoke its ADPM-RESET.request primitive to reset itself.

Figure 36 describes the messages exchanged during the removal of a device initiated by the device itself.

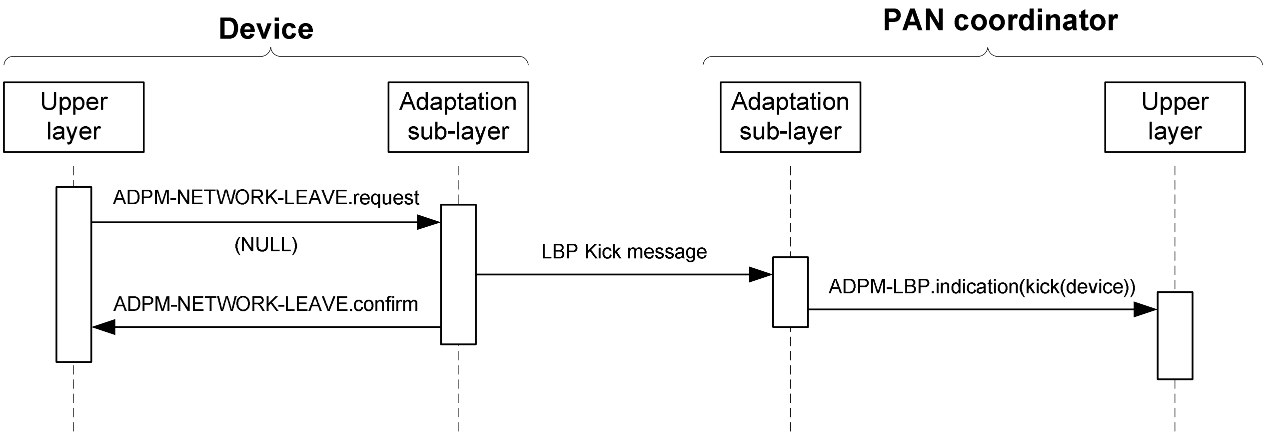


Figure 36 – Message sequence chart during removal of a device by itself

On the PAN coordinator side, an ADPM-LBP.indication containing the KICK message received is generated to inform the upper layers. This message contains the 64-bit address of the device which removed itself from the PAN

7.4.6 Fragment Recovery (based on draft-thubert-6lowpan-simple-fragment-recovery-02)

The fragment recovery as described in draft-thubert-6lowpan-simple-fragment-recovery-02 applies, with the selections specified in Table 57.

Table 57 – Selections from draft-thubert-6lowpan-simple-fragment-recovery-02

Clause	Title & remarks/modifications	Statement
1.	Introduction	N
2.	Terminology	N
3.	Rationale	N
4.	Requirements	N
5.	Overview	N
6.	New Dispatch types and headers	N

Clause	Title & remarks/modifications	Statement
6.1.	Recoverable Fragment Dispatch type and Header	N
6.2.	Fragment Acknowledgement Dispatch type and Header	N
7.	Outstanding Fragments Control	N
8.	Security Considerations	N
9.	IANA Considerations	N
10.	Acknowledgments	N/R
11.	References	N
11.1.	Normative References	N
11.2.	Informative References	I

7.4.7 Spy Mode

This mode is used to have a spy modem supervising all transmission on its behaviour. Once activated; the modem will process all packets like its own. The spy modem will generate an ADPD-DATA.indication for all packets received. It must also, behave like in normal mode; processing and forwarding packets.

For example, on reception of an MCPS-DATA.indication which is not mine, the modem, in this case, will generate an ADPD-DATA.indication for upper layer then forward the packet toward the destination.

The route cost of all links with a spy modem is set to 31 so we prevent routing a packet via such modem.

If a spy modem receives a fragment, it will add an IPv6 fragment header to the packet so the upper layer can detect it. The fragment offset field will be set to the offset of the LOWPAN header and the Identification field to the Datagram_Tag.

7.5 Functional description

7.5.1 Network formation

The network formation can only be performed by the PAN coordinator. Any device other than the PAN coordinator must not attempt to perform a network formation.

Prior to the network formation, the PAN coordinator must perform an active scan as described in 7.4.5.2.2.2. If the PANDescriptorList given by the ADPM-DISCOVERY.confirm primitive is empty, then the PAN coordinator can start a new network. If the PANDescriptorList is not empty, the PAN coordinator should inform the rest of the system that a PAN is already operating in the POS of the device, and may start a new network afterwards. The procedures and decisions associated with this behaviour are implementation specific.

After the network discovery, the PAN coordinator must set its PAN ID to the predefined value stored in it. This value can be obtained remotely from a configuration server, or locally computed. The way how this PAN ID is chosen and set in the coordinator is implementation specific.

NOTE The PAN identifier must be logically ANDed with 0xFCFF, as described in 7.4.3 of the present document (clause 6 of RFC 4944).

Once the PAN identifier has been determined, the adaptation sublayer must invoke the MLME-START.request with the following parameters:

- PANId = the PAN identifier computed;
- LogicalChannel = 0 (not used);
- ChannelPage = 0 (not used);

- StartTime = 0 (not used);
- BeaconOrder = 15 (beaconless network);
- SuperframeOrder = 15 (not used);
- PANCoordinator = TRUE;
- BatteryLifeExtension = FALSE (not used);
- CoordRealignment = FALSE;
- CoordRealignSecurityLevel, CoordRealignKeyIdMode, CoordRealignKeySource and CoordRealignKeyIndex: not used, should be set to 0;
- BeaconSecurityLevel = 0;
- BeaconKeyIdMode, BeaconKeySource, BeaconKeyIndex: not used, should be set to 0.

The MAC sublayer then generates a MLME-START.confirm primitive with the corresponding status code, which is forwarded to the upper layers through the generation of an ADPM-NETWORK-START.confirm.

7.5.2 PAN ID conflict detection and handling

At any time, when a device is associated to a PAN, its MAC sublayer must analyze the destination and source PAN Identifier in the MAC header of any frame it receives.

If a frame containing a destination or source PAN Identifier is received and does not match the PAN Identifier of the device, it must generate a MLME-SYNC-LOSS.indication primitive with the following characteristics:

- LossReason = PAN_ID_CONFLICT;
- PANId = The conflicting PAN ID;
- LogicalChannel = 0 (not used);
- ChannelPage = 0 (not used);
- SecurityLevel = 0 (not used);
- KeyIdMode, KeySource and KeyIndex can be ignored.

If the adaptation sublayer receives a MLME-SYNC-LOSS.indication primitive with another LossReason than PAN_ID_CONFLICT, it must ignore it.

In response, the adaptation layer must generate a CONFLICT frame to its PAN coordinator. This frame is a standard LBP message frame with its Code field set to 101b. The bootstrapping data in that message should contain the PAN Id of the detected PAN using the format defined in clause 3.3.1 of draft-6lowpan-commissioning-02 and described in Figure 37:

Bits: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Attribute ID	1	1	0 0 0 0 0 0 1 0	Conflicting PAN Id

Figure 37 – CONFLICT message format

This frame is sent using an ADPD-DATA.request primitive with the following attributes:

- DstAddrMode = 0x02;
- DstAddr = IPv6 destination address, formed with the short address of 0x0000;
- NsduLength = the length of the frame;

- Nsdu = the frame;
- NsduHandle = a random number;
- MaxHops = adpMaxHops;
- DiscoverRoute = TRUE;
- QualityOfService = FALSE;
- SecurityEnabled = TRUE.

A device must wait adpPANConflictWait seconds between two consecutive sending of a CONFLICT frame for the same conflicting PAN Id, and the total number of CONFLICT frames sent for a given conflicting PAN Id must not exceed adpMaxPANConflictCount. When this value is reached, the device must stop sending CONFLICT frames for this conflicting PAN Id.

When the PAN coordinator receives this frame, it must generate an ADPM-NETWORK-STATUS.indication primitive to the upper layer, with:

- The Status field sets to PAN_ID_CONFLICT; and
- The AdditionalInformation field sets to the conflicting PAN Id.

8 Security

8.1 Access control and authentication

An End Device (ED) may not access to the network without a preliminary Identification (with comparison to white or black lists) and Authentication. Identification and Authentication are based on two parameters that personalized every ED:

- A EUI-48 MAC address as defined in IEEE 802. This address may be easily converted into a EUI-64 as required by IEEE 802.15.4 and related documents;
- A 128-bit shared secret (aka Pre-Shared Key or PSK) used as a credential during the authentication process. It is shared by the ED itself (aka peer) and an authentication server. The mutual authentication is based on a proof the other party knows the PSK. It is of highest importance, the PSK remains secret.

The Identification and Authentication processes are activated when an ED restarts and may also be launched at any time according to the security policy in place. The related material is carried by the 6LoWPAN Bootstrapping Protocol (LBP) (see 7.4.5) that embeds the Extensible Authentication Protocol (EAP) (see 7.4.5.2.1.2).

As shown in Figure 38, LBP and EAP have been designed to be relayed by intermediates nodes. Then during the Bootstrapping phase, when an ED (aka LBD) that have not yet acquired a routable 16-bit address, is a 1-hop distance of the PAN Coordinator (aka LBS) they can directly communicate. Otherwise, they must use an intermediate node (aka LBA) located at 1-hop distance of the LBD.

Moreover, two different authentication architectures must be considered:

- The authentication server function is directly supported by the LBS, and in this case all the authentication material (access lists, credentials, etc.) must be loaded in the LBS;
- The authentication server function is supported by a remote (and usually centralized) AAA server, and in this case, the LBS is only in charge of forwarding the EAP messages to the AAA server over a standard AAA protocol (i.e. RADIUS RFC 2865).

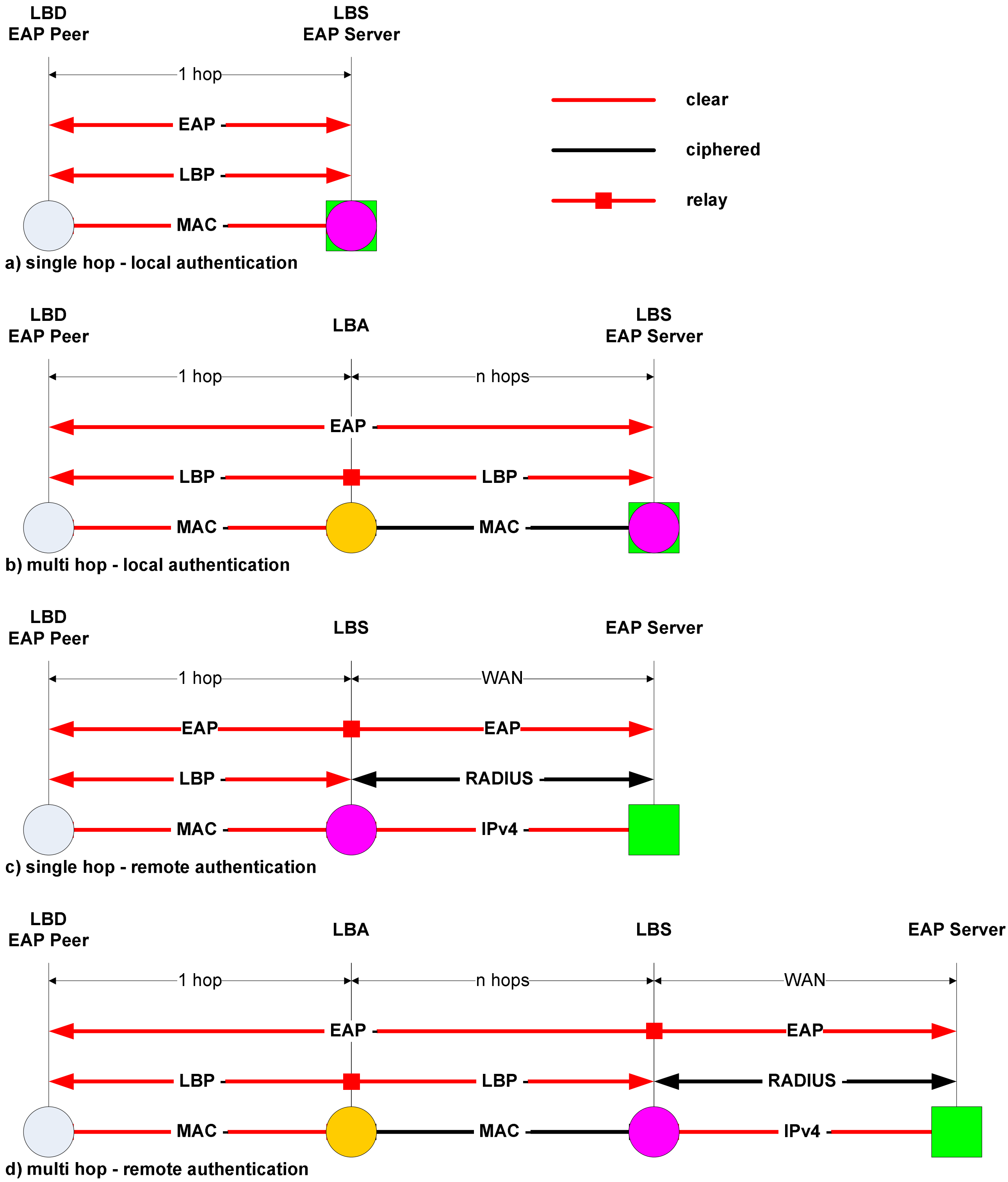


Figure 38 – LBP and EAP Relaying Capabilities

The Authentication process is fully dependant of the EAP method in place. The EAP protocol is very flexible and support various EAP methods (EAP-MD5, EAP-AKA, EAP-TLS, etc.). Each method is characterized by its credentials (shared secret, certificate, SIM cards, etc.) and by its signature and encryption algorithms.

The method adopted for the OFDM CPL network is EAP-PSK (see 7.4.5), the main design goals of which are:

- Simplicity: it is entirely based on a single credential (a 128-bit Pre-Shared Key) and a single cryptographic algorithm (AES-128);
- Security: it appears very conservative in its design following well-known and improved cryptographic schemes.;
- Extensibility: in the OFDM CPL case, it is easily extended to support Group Key distribution (see 8.5.2).

8.2 Confidentiality and integrity

As shown by Figure 39, confidentiality and integrity services are ensured at different levels:

- At MAC level: as defined in IEEE 802.15.4, a CCM* type of ciphering is delivered to every frame transmitted between nodes in the network. It's a universal Low Layer Confidentiality and Integrity service (with anti-replay capabilities). The MAC frames are encrypted and decrypted at every hop. The only exceptions are some well-controlled frames in the early stages of the Bootstrapping process. To fairly support this service, all the nodes in the network receive the same Group session key (GMK). This GMK is individually and securely distributed to every node by using the EAP-PSK Secure Channel;

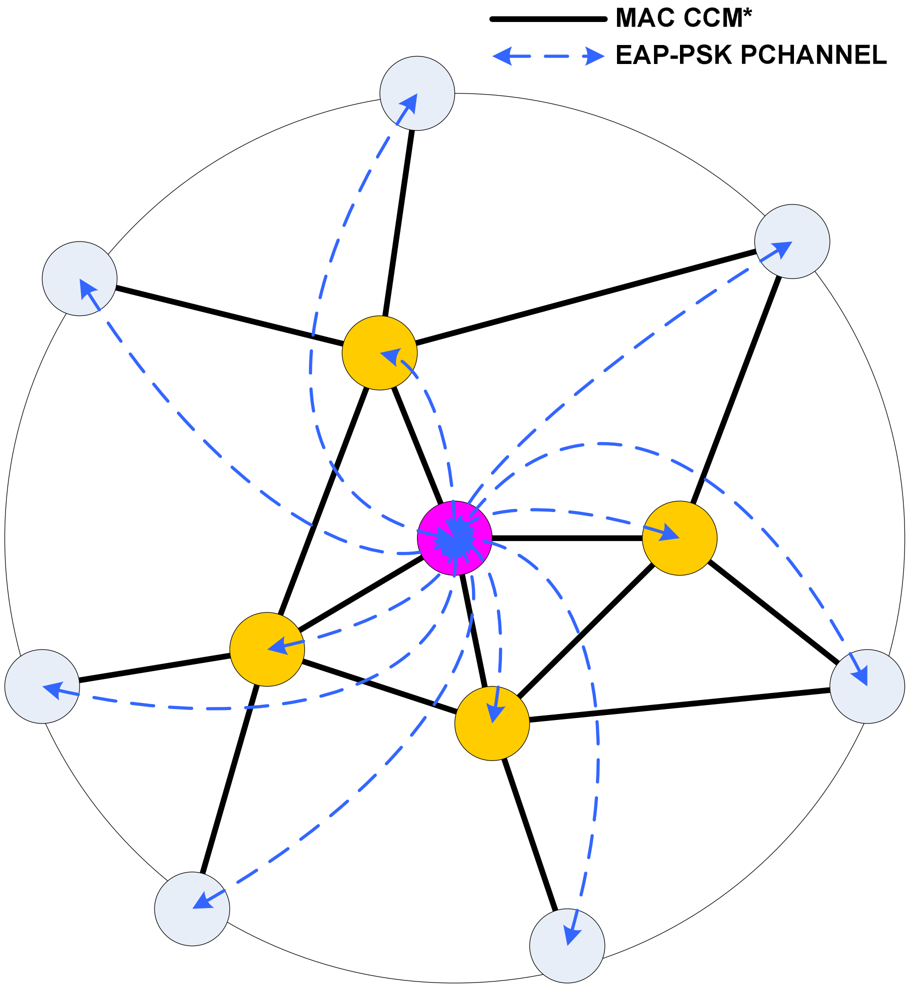


Figure 39 – Confidentiality and Security

- At the EAP-PSK level: as defined in RFC 4764, EAP-PSK provides Confidentiality and Integrity (and Replay Protection) services, also known as Protected Channel (PCHANNEL) to the messages exchanged over EAP between the EAP server and any peer.

8.3 Anti-Replay and DoS prevention

It is always difficult to prevent DoS attacks, and especially those targeting the Physical level, but by nature their impact is limited to a small area.

The CCM* ciphering mode is generalized at MAC layer. It prevents unauthenticated devices accessing the network and having malicious actions on routing, provisioning and any other Low Layer processes. The only exception is the well-controlled Bootstrapping process.

Moreover, an anti-replay mechanism is specified at the MAC sublayer.

8.4 Authentication and key distribution protocol – Selections from RFC 3748

Authentication and key distribution are supported by the Extensible Authentication Protocol (EAP) as given in RFC 3748 together with the selections listed in Table 58.

Table 58 – Selections from RFC 3748

Clause	Title & remarks/modifications	Statement
1.	Introduction	N
2.	Extensible Authentication Protocol (EAP) - Initial Identity Request (allow roaming and EAP method negotiation) is let for further study and must be bypassed.	S
2.1	Support for Sequences	N
2.2	EAP Multiplexing Model - Only one EAP method is defined (cf. 7.4.5)	S
2.3	Pass-Through Behaviour - Over LBP, the Code field is slightly different from a regular EAP Code field as specified in RFC 3748. The conversion appears straightforward in both directions. The proper conversion must apply when the EAP message is propagated over another protocol (i.e. RADIUS) and in case of integrity protection covering the EAP header	S
2.4	Peer-to-Peer Operation	N
3.	Lower Layer Behaviour	N
3.1	Lower Layer Requirements - LBP and underlying protocols provide: - Reliable transport - Error detection (CRC) - No Lower Layer security when bootstrapping - MTU size greater than 1 020 octets (by fragmentation) - No duplication - Ordering guaranties	S
3.2	EAP Usage Within PPP	N/R
3.3	EAP Usage Within IEEE802	N/R
3.4	Lower Layer Indications	N
4.	EAP Packet Format - Over LBP, the Code field is slightly different from a regular EAP Code field.	S
4.1	Request and Response - Over LBP, the Code field is slightly different from a regular EAP Code field.	S
4.2	Success and Failure - Over LBP, the Code field is slightly different from a regular EAP Code field.	S
4.3	Retransmission Behaviour	N

Clause	Title & remarks/modifications	Statement
5.	Initial EAP Request / Response Types - For the Type field, the only available values are 3 (Nak – in Response only) and the value assigned to the EAP method (see 8.5). Other values are left for further study	S
5.1.	Identity	N/R
5.2.	Notification	N/R
5.3.	Nak	N
5.4.	MD5-Challenge	N/R
5.5.	One-Time Password (OTP)	N/R
5.6.	Generic Token Card (GTC)	N/R
5.7.	Expanded Types	N/R
5.8.	Experimental	N/R
6.	IANA Considerations	N
7.	Security Considerations	N
8.	Acknowledgements	I
9.	References	N
Appendix A.	Changes from RFC2284	I

8.5 EAP Method

The EAP protocol is very flexible and support various EAP methods (EAP-MD5, EAP-AKA, EAP-TLS, etc.). Each method is characterized by its credentials (shared secret, certificate, SIM cards, etc.) and by its signature and encryption algorithms.

For the OFDM CPL case, the recommended method is Pre-Shared Key EAP Method (EAP-PSK) as given in RFC 4764 together with the selections listed in Table 59.

Table 59 – Selections from RFC 4764

Clause	Title & remarks/modifications	Statement
1.	Introduction	N
2.	Protocol Overview	N
3.	Cryptographic Design of EAP-PSK	N
4.	EAP-PSK Message Flows - EAP-PSK extension capabilities are used for Group Key distribution in full compliance to RFC 4764. See 8.5.2	N
5.	EAP-PSK Message Format - EAP-PSK extension capabilities are used for Group Key distribution in full compliance to RFC 4764. See 8.5.2	N
6.	Rules of Operation for EAP-PSK Protected Channel	N
7.	IANA Considerations	N
8.	Security Considerations	N
9.	Security Claims	I
10.	Acknowledgements	I
11.	References	N
Appendix A.	Generation of the PSK from a Password - Discouraged	N/R

8.5.1 Overview of EAP-PSK

EAP-PSK, according to the EAP specification supports the following key hierarchy:

Pre-Shared Key (PSK)	PSK is the long-term 128-bit credential shared by the EAP server and the peer
Authentication Key (AK)	A 128-bit key derived from the PSK that the EAP peer and server use to mutually authenticate
Key-Derivation Key (KDK)	A 128-bit key derived from the PSK that the EAP peer and server use to derive session keys (such as TEK, MSK and EMSK)
Transient EAP Key (TEK)	A session key that is used to establish a protected channel between the EAP peer and server during the EAP authentication. EAP-PSK uses a 128-bit TEK in conjunction with AES-128 in EAX mode of operation as a cipher suite.
Master Session Key (MSK)	A session key derived between the EAP peer and server. EAP-PSK generates a 512-bit MSK that may be used to provide security at the Application level.
Extended Master Session Key (EMSK)	A session key derived between the EAP peer and server. EAP-PSK generates a 512-bit EMSK. It is not used in OFDM CPL and must not be generated.

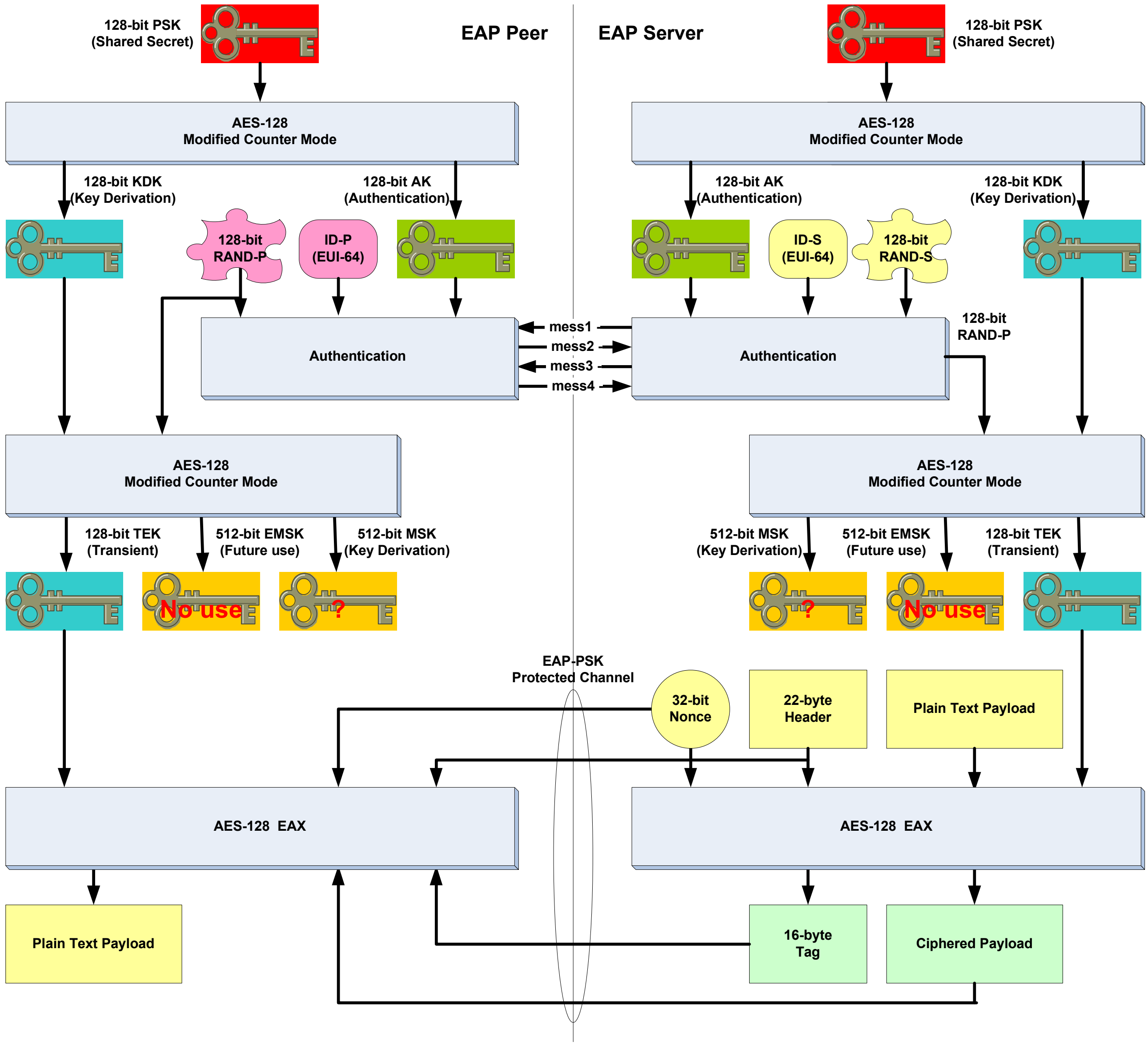


Figure 40 – EAP-PSK Key Hierarchy overview

8.5.2 Group Key distribution

The 128-bit Group Master Key (GMK) is generated by the EAP Server. Then it is securely and individually delivered to the EAP peers via the EAP-PSK Protected Channel (PCHANNEL).

GMK is assumed being random. GMK generation is considered as purely implementation dependant.

GMK is distributed to the peer in two circumstances:

- During the Bootstrapping process, carried as a regular extension to EAP-PSK message 3 of the Figure 40;
- During the Re-keying process, carried as a regular extension to EAP-PSK message 5 of the Figure 40. The GMK lifetime is rather long (several 10s years) due to the 4 byte counter included in the nonce. Nevertheless it's of good policy to timely re-key the network or when a node is leaving it.

8.5.3 GMK field format

The GMK field in message 3 or 5 of the Figure 40 is defined in compliance with the generic extension field (EXT) (see RFC 4764 clause 5.3.) described in Figure 41.

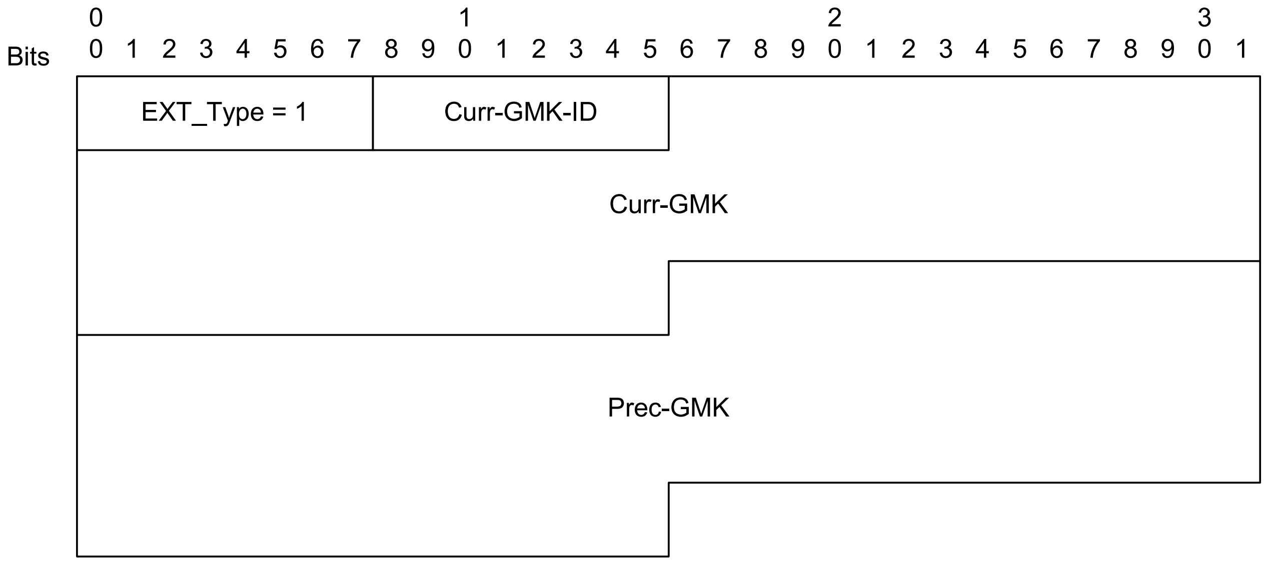


Figure 41 – GMK field format

where

EXT_Type The EXT_TYPE field is one octet and indicates the type of the Extension

1 GMK

Curr-GMK-ID The Curr-GMK-ID field is one octet and represents the Key Identifier of the current GMK.

Curr-GMK The Curr-GMK is 16 octets and contains the value of the current GMK.

Prec-GMK The Prec-GMK is 16 octets and contains the value of the preceding GMK.

8.5.4 Peer side procedure

When a peer receives this field embedded in a message 3 in case of Bootstrapping, or a message 5 in case of Re-keying (refer to Figure 40), it installs both keys in their respective slot and process the message according to RFC 4764.

In every case, both keys are immediately available in reception according to the Key Identifier contained in the MAC header of the received frame.

In case of Bootstrapping, the peer keeps sending the frames in clear text up to the reception of an EAP Success message. Then, it starts sending ciphered frames using the current GMK.

In case of Re-keying, the peer keeps sending messages according to the previously assigned policy until reception of an EAP Success message. Then, it starts sending frames using the current GMK.

After switching to the current GMK, a peer may keep receiving some messages encrypted with the preceding GMK during a transient period. The previous GMK must be deleted after a configurable delay (default value = 10 min).

8.5.5 Server side procedure

The Bootstrapping procedure is defined in 7.4.5.2.2.

In case of re-keying, the EAP server generates a new GMK. Then it transmits a LBP challenge message, embedding an EAP Request message that contains the newly generated GMK, coupled with the preceding one, to every formerly authenticated peer. Upon reception of the corresponding EAP Response, or after a configurable delay (default value = 10 min), the server starts sending EAP Success messages for the validation of the new GMK.

Annex A
(normative)

Interleaver pattern generator

The following piece of code is used for generating a good interleaving pattern. It needs two parameters, freqNum which is the number of data-holding sub-carriers, and symbNum which is the number of OFDM symbols. The interleaving table will be generated in ILV_TBL array.

```
void Interleaver_init( int freqNum, int symbNum )
{
    volatile int i, j, I, J, m, n, m_i, m_j, n_i, n_j;
    n = symbNum;
    m = freqNum;
    n_j = 1; n_i = 1;
    m_i = 1; m_j = 1;

    for ( i = 3; i < n; i++ )
    if ( gcd(n,i) == 1 )
    {
        n_j = i;
        break;
    }
    for ( i++; i < n; i++ )
    if ( gcd(n,i) == 1 )
    {
        n_i = i;
        break;
    }
    for ( i = 3; i < m; i++ )
    if ( gcd(m,i) == 1 )
    {
        m_i = i;
        break;
    }
    for ( i++; i < m; i++ )
    if ( gcd(m,i) == 1 )
    {
        m_j = i;
        break;
    }

    ILV_SIZE = m * n

    for ( j = 0; j < n; j++ )
```

```
        for ( i = 0; i < m; i++ )
        {
                J = ( j * n_j + i * n_i )  % n;
                I = ( i * m_i + J * m_j )  % m;
                ILV_TBL[ i + j * m ] = I + J * m;
        }
}
```

Interleaving itself can be done using the following piece of code:

```
    for ( i = 0; i < size; i += ILV_SIZE )
    for ( j = 0; j < ILV_SIZE; j++ )
       y[ i + ILV_TBL[j] ] = (i+j) < size ? x[i+j] : 0;
```

Similarly, de-interleaving can be done using

```
    for ( i = 0; i < size; i += ILV_SIZE )
    for ( j = 0; j < DLV_SIZE; j++ )
       y[i+j] = x[ i + ILV_TBL[j] ];
```

Annex B
(normative)

Protocol Implementation Conformance Statement

B.1 Overview

The first part of this annex entirely takes as reference the protocol implementation conformance statement of the IEEE 802.15.4, annex D.

The second part of this annex gives similar tables to ensure that all items related to the physical layer of PLC OFDM Type 2 have been taken into account.

B.2 PICS proforma tables

B.2.1 Functional device types (from annex D.7.1 of IEEE 802.15.4)

Table B.1 – PICS – Functional device types (from annex D.7.1 of IEEE 802.15.4)

Item number	Support			Comments
	N/A	Yes	No	
FD1		X		
FD2			X	
FD3		X		
FD4		X		
FD5		X		

B.2.2 PHY functions (from annex D.7.2.1 of IEEE 802.15.4)

Table B.2 – PICS – PHY functions (from annex D.7.2.1 of IEEE 802.15.4)

Item number	Support			Comments
	N/A	Yes	No	
PLF1		X		
PLF2		X		
PLF3	X			Radio specific requirement
PLF4	X			Radio specific requirement
PLF5	X			Radio specific requirement
PLF6		X		
PLF7	X			Radio specific requirement
PLF8		X		
PLF8.1	X			Radio specific requirement
PLF8.2		X		
PLF8.3	X			Radio specific requirement

B.2.3 PHY packet (from annex D.7.2.2 of IEEE 802.15.4)

Table B.3 – PICS – PHY packet (from annex D.7.2.2 of IEEE 802.15.4)

Item number	Support			Comments
	N/A	Yes	No	
PLP1		X		

B.2.4 Radio frequency (from annex D.7.2.3 of IEEE 802.15.4)

Table B.4 – PICS – Radio frequency (from annex D.7.2.3 of IEEE 802.15.4)

Item number	Support			Comments
	N/A	Yes	No	
RF1	X			Radio specific requirement
RF1.1	X			Radio specific requirement
RF1.2	X			Radio specific requirement
RF1.3	X			Radio specific requirement
RF1.4	X			Radio specific requirement
RF2	X			Radio specific requirement

B.2.5 MAC sublayer functions (from annex D.7.3.1 of IEEE 802.15.4)

Table B.5 – PICS – MAC sublayer functions (from annex D.7.3.1 of IEEE 802.15.4)

Item number	Support			Comments
	N/A	Yes	No	
MLF1		X		
MLF1.1			X	Indirect transmission is not supported
MLF2		X		
MLF2.1		X		
MLF2.2		X		
MLF2.3		X		
MLF3		X		
MLF3.1		X		
MLF3.2		X		
MLF4		X		
MLF5			X	
MLF5.1			X	
MLF5.2			X	
MLF6		X		
MLF7		X		
MLF8			X	Performed by 6LoWPAN
MLF9		X		
MLF9.1		X		
MLF9.2		X		
MLF9.2.1		X		

Item number	Support			Comments
	N/A	Yes	No	
MLF9.2.2		X		
MLF10.1	X			Radio specific requirement
MLF10.2		X		
MLF10.3			X	Not necessary for non beacon-enabled networks
MLF10.4			X	
MLF11			X	
MLF12			X	
MLF13			X	

B.2.6 MAC frames (from annex D.7.3.2 of IEEE 802.15.4)

Table B.6 – PICS – MAC frames (from annex D.7.3.2 of IEEE 802.15.4)

Item number	Support						Comments
	Transmitter			Receiver			
	N/A	Yes	No	N/A	Yes	No	
MF1		X			X		
MF2		X			X		
MF3		X			X		Acknowledgement frames are described in PHY specification associated with the present specification and annex 6 of the present document
MF4		X			X		
MF4.1			X			X	Association performed by 6LoWPAN
MF4.2			X			X	Association performed by 6LoWPAN
MF4.3			X			X	Association performed by 6LoWPAN
MF4.4			X			X	No transaction support
MF4.5			X			X	Performed by 6LoWPAN
MF4.6			X			X	
MF4.7		X			X		
MF4.8			X			X	
MF4.9			X			X	

B.3 Conformance to PLC OFDM Type 2 physical layer

NOTE Item numbers in Table B.7 refer to subsections of section 6 Physical layer specification.

Table B.7 – conformance to PLC OFDM Type 2 physical layer

Item number	Support			Comments
	N/A	Yes	No	
6.2.2		X		
6.2.3		X		
6.2.4		X		

Item number	Support			Comments
	N/A	Yes	No	
6.2.5.1		X		
6.2.5.2		X		
6.2.6		X		
6.3.1.2		X		
6.3.1.3		X		
6.3.1.4		X		
6.3.2		X		
6.3.3		X		
6.3.4		X		
6.6.1		X		
6.6.2		X		
6.6.3		X		
6.6.4		X		
6.6.5		X		
6.7.1		X		
6.7.2		X		
6.8.1		X		
6.8.2		X		
6.9		X		
6.10.1		X		
6.10.2		X		
6.10.3		X		
6.10.4		X		
6.10.5		X		
6.11.1		X		All primitives defined in the related subsections must be implemented
6.11.2		X		All primitives defined in the related subsections must be implemented

2139

Annex C
(informative)

Routing Cost

This part describes the characteristics a routing cost used in the LOAD routing algorithm (described in draft-daniel-6lowpan-load-adhoc-routing-03 and in 7.4.4) must have.

A route cost is defined as the sum of all the link costs on the route. As described in draft-daniel-6lowpan-load-adhoc-routing-03, a route cost is an integer value between 0 and 255, lower values meaning better routes.

As there can be at most 8 hops on a route as defined in 7.4.3 (see clause 5.2 of RFC 4944), a link cost is an integer between 0 and 31.

If we note P a route which goes through devices $\{D_0, D_1, ..., D_{N-1}\}$, where N is the number of hops on the route ($0 < N \leq 8$), and $C\{[D_i, D_j]\}$ the link cost between devices D_i and D_j, the route cost $RC(P)$ of P can then be defined as:

$$RC(P) = \sum_{i=0}^{N-1} C\{[D_i, D_{i+1}]\}$$

The link cost should take into account PHY transmission parameters, number of hops, etc… The link cost computation algorithm is implementation dependant.

Annex D
(normative)

Channel access

D.1 Overview

The channel access is accomplished by using the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism with a random backoff time. The random backoff mechanism spreads the time over which stations attempt to transmit, thereby reducing the probability of collision. Each time a device wishes to transmit data frames, it shall wait for a random period. If the channel is found to be idle, following the random backoff, the device shall transmit its data. If the channel is found to be busy, following the random backoff, the device shall wait for another random period before trying to access the channel again.

A Carrier sense is a fundamental part of the distributed access procedure. Physical Carrier Sense (PCS) is provided by the PHY upon detection of the Preamble. In the latter case, PCS shall stay high long enough to be detected and Virtual Carrier Sense (VCS) to be asserted by the MAC. A virtual carrier sense mechanism is provided by the MAC by tracking the expected duration of channel occupancy. Virtual carrier sense is set by the length of received packet or upon collision. In these cases, virtual carrier sense tracks the expected duration of the Busy state of the medium. The medium shall also be considered Busy when the station is transmitting.

A VCS timer is maintained by all stations to improve reliability of channel access. The VCS timer is set based on received long (data) or short (ACK) frames. The VCS timer is also set upon collision or when the station powers up. Stations use this information to compute the expected Busy condition of the medium or the expected duration of the Contention State and store this information in the VCS timer.

A Collision occurs in each of the following circumstances:

– The transmitting station receives a something other than ACK or NACK response when a response is expected.
– The transmitting station shall infer a Collision from the absence of any response to a transmission when a response is expected. Note that the absence of a response could also be the result of a bad channel. Since there is no way to distinguish between the two causes a Collision is inferred.

D.2 Interframe (IFS) Spacing

A time intervals between frames on the medium constitute the Interframe Space and are necessary due to propagation and processing time. As shown in Figure D.1, three interframe space values are defined. Contention Interframe Space (CIFS) occurs after the end of the previous transmission. The second defined interval is the Response Interframe Space (RIFS).

RIFS is the time between the end of a transmission and the start of its associated response. If no response is expected, the CIFS is in effect.

An Extended Interframe Space (EIFS) is defined for conditions when the station does not have complete knowledge of the state of the medium. This can occur when the station initially attaches to the network, when errors in the received frames make them impossible to decode unambiguously. If a packet is received and correctly decoded before the expiration of the EIFS, then the EIFS is cancelled. The EIFS is significantly longer than the other interframe spaces, providing protection from Collision for an ongoing frame transmission or segment burst when any of these conditions occur. The EIFS is calculated as follows:

aEIFS = aAckTime + aCIFS + aRIFS + MaxFrameSize*aSymbolTime

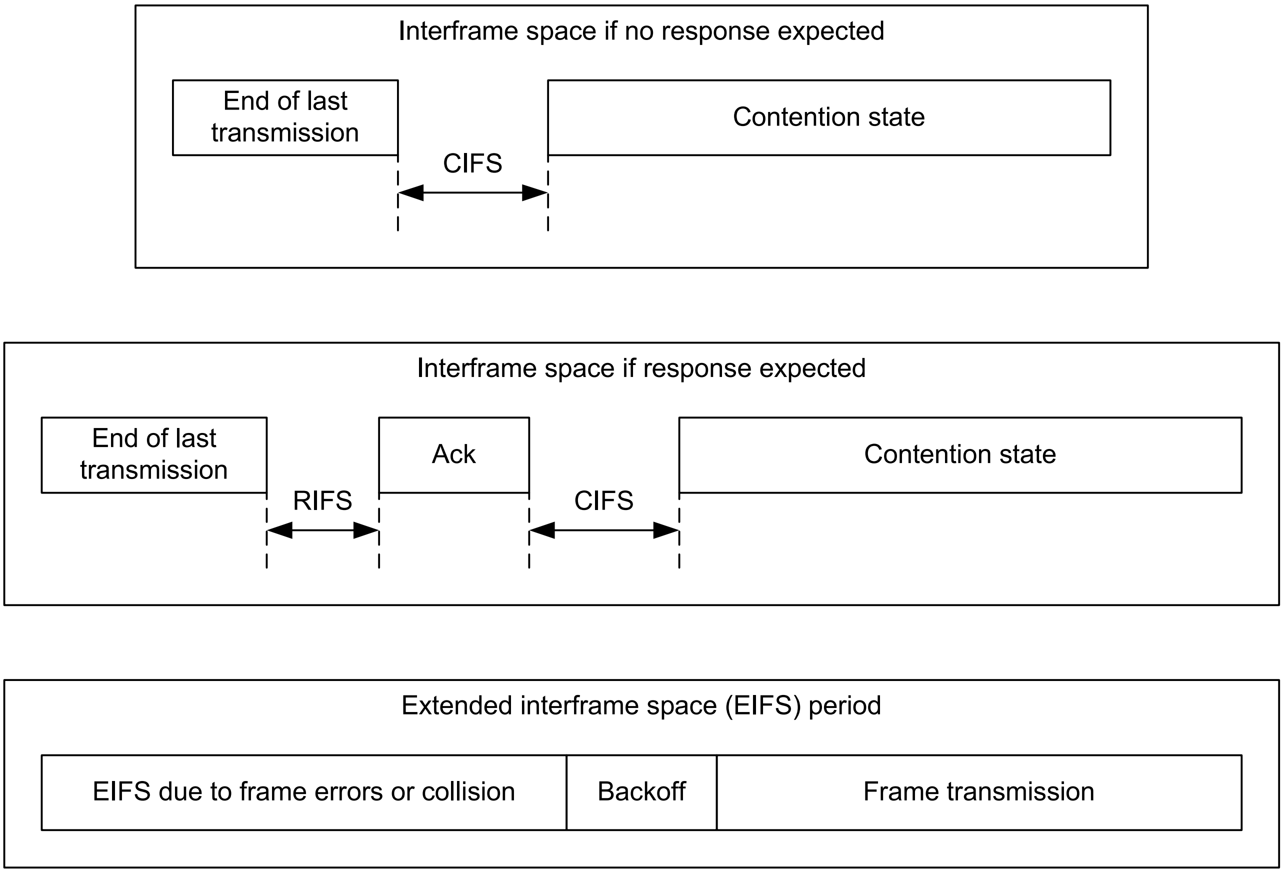


Figure D.1 – IFS

D.3 CSMA-CA

The present specification supports only an unslotted version of the CSMA-CA algorithm for non-beacon PAN described in IEEE 802.15.4

The random backoff mechanism spreads the time over which stations attempt to transmit, thereby reducing the probability of collision, using a truncated binary exponential backoff mechanism.

The CSMA-CA algorithm shall be used before the transmission of data or MAC command frames

The algorithm is implemented using units of time called backoff periods, where one backoff period shall be equal to u*nitBackoffPeriod* symbols.

Each device shall maintain two variables for each transmission attempt: ***N*B** and ***B*E**. ***NB*** is the number of times the CSMA-CA algorithm was required to backoff while attempting the current transmission; this value shall be initialized to 0 before each new transmission attempt.

BE is the backoff exponent, which is related to how many backoff periods a device shall wait before attempting to assess a channel. ***BE*** shall be initialized to the value of ***minB*E**.

Note that if ***minBE*** is set to 0, collision avoidance will be disabled during the first iteration of this algorithm. Figure D.2 illustrates the steps of the CSMA-CA algorithm. The MAC sublayer shall first initialize ***NB**,* and ***BE*** [step (1)] and then proceed directly to step (2).

The MAC sublayer shall delay for a random number of complete backoff periods in the range 0 to 2^{BE} –1 [step (2)] and then request that the PHY perform a PCS (Physical Carrier Sense) [step (3)].

$$\text{Backoff Time} = \text{Random}(2^{BE} - 1) \times \text{aSlotTime}$$

If the channel is assessed to be busy [step (4)], the MAC sublayer shall increment both ***NB*** and ***BE*** by one, ensuring that ***BE*** shall be no more than ***maxB*****E**. Note: for high priority packets ***maxB*****E** should be equal to ***minBE.***

If the value of ***NB*** is less than or equal to ***maxCSMABackoff*****s**, the CSMA-CA algorithm shall return to step (2).

If the value of ***NB*** is greater than ***maxCSMABackoff*****s**, the CSMA-CA algorithm shall terminate with a Channel Access Failure status.

If the channel is assessed to be idle [step (5)], the MAC sublayer shall begin transmission of the frame immediately.

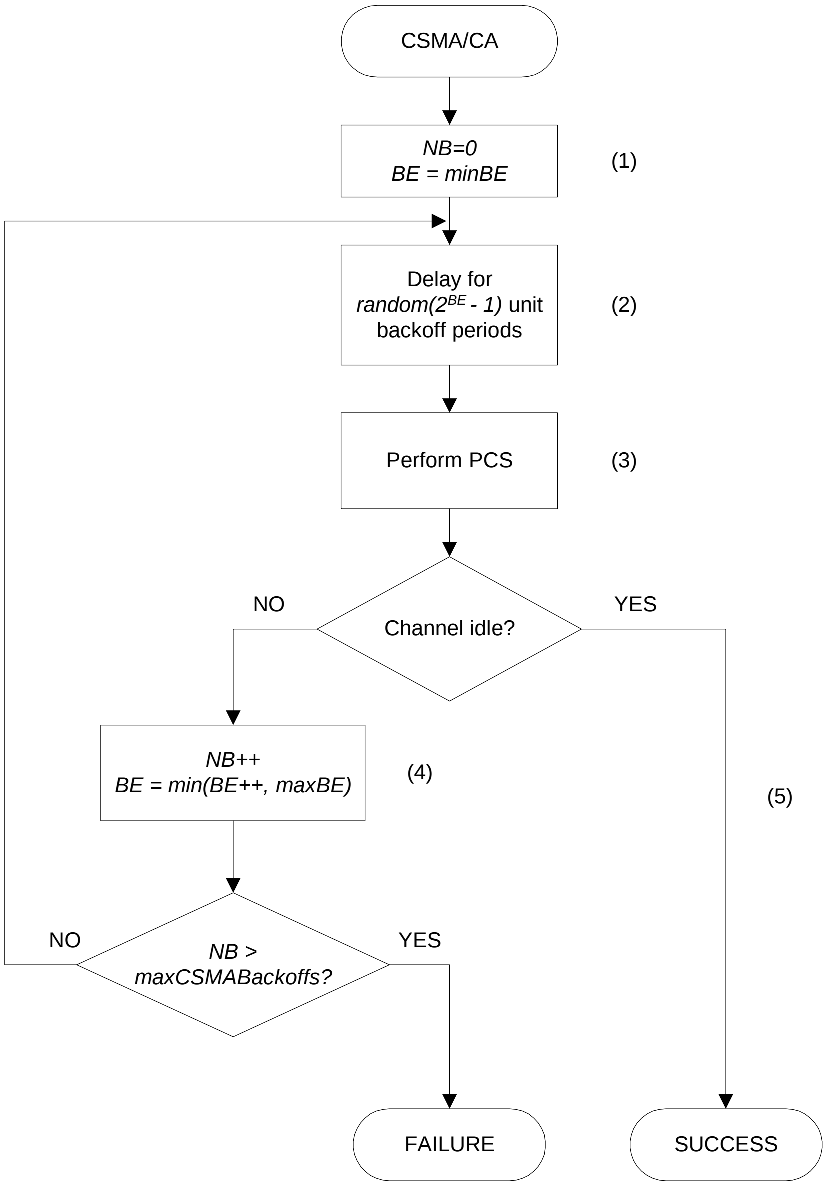


Figure D.2 – CSMA/CA algorithm

D.4 Priority

Prioritized access to the channel can be beneficial for real time application or control application when urgent message should be delivered as soon as possible. Only two levels of priority (High and Normal) will be used to minimize complexity. Priority resolution is implemented by using two contention time windows during contention state as shown in Figure D.3:

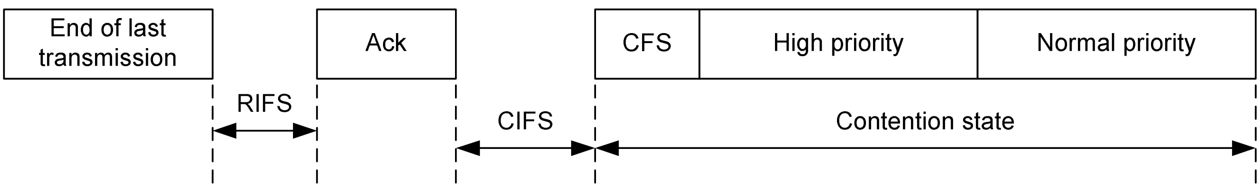


Figure D.3 – Priority Contention Windows

First slot of contention window is called Contention Free Slot (CFS). It is used to implement packet bursting without backoff procedure in order to prevent possible interruption from other nodes.

The high and normal priority stations will compete for channel during HPCW and NPCW correspondingly. Since HPCW is located before NPCW high priority stations will get access to the channel before station with normal priority. Duration of HPCW and NPCW are calculated as follow:

$$\text{HPCW time} = \textit{macHighPrioirtyWindowSize} * \textit{aSlotTime;}$$

$$\text{NPCW time} = (2^{maxBE} * \textit{aSlotTime}) - \text{HPCW time;}$$

$$\text{CFS time} = \textit{aSlotTime};$$

D.5 ARQ

ARQ (Automatic Repeat reQuest) is implemented based on acknowledged and unacknowledged retransmission. The MAC sublayer uses a response type as part of its ARQ mechanism. ACK is a traditional positive acknowledgment that when received allows the transmitter to assume successful delivery of the frame. The negative acknowledgment (NACK) is used to inform a packet originator that the receiver received the packet but it was corrupted.

A successful reception and validation of a data can be confirmed with an acknowledgment. If the receiving device is unable to handle the received data frame for any reason, the message is not acknowledged.

If the originator does not receive an acknowledgment after waiting period, it assumes that the transmission was unsuccessful and retries the frame transmission. If an acknowledgment is still not received after several retries, the originator can choose either to terminate the transaction or to try again. When the acknowledgment is not required, the originator assumes the transmission was successful. Also if acknowledgment is not required, the originator can retransmit the same packets few times to increase probability of data delivery. The receiver should be able distinguish and discard redundant copies using the Sequence Number and Segment Count. The retransmitted packet will have the same Sequence Number and Segment Count as original.

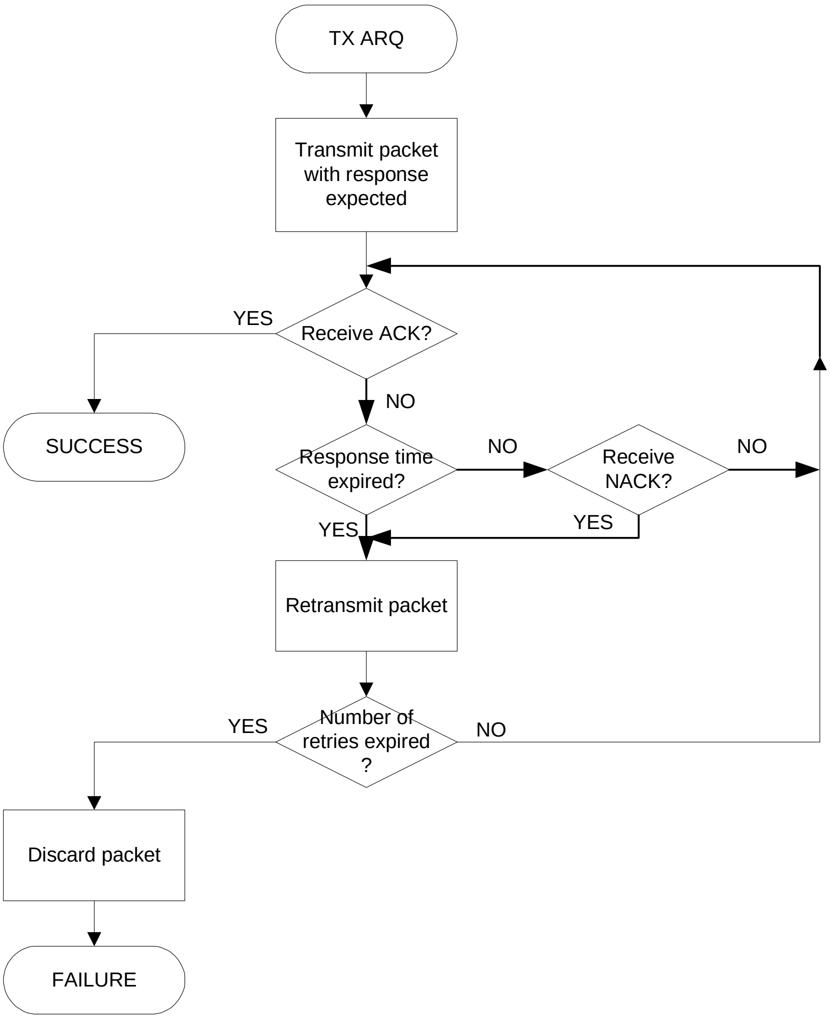
The acknowledgment cannot be requested for broadcast or multicast transmission. On transmit side ARQ requires configurable number of retransmissions (cf. macMaxFrameRetries from 7.4.2 of IEEE 802.15.4) as shown in Figure D.4.

On receive side ARQ generates acknowledgement for PLC packet with correct FCS (CRC16) if packet corresponds to this address as shown in Figure D.5.

The received packet FCS (16 bit) will be sent back to the packet originator as a part of an acknowledgement (Frame Control Header).

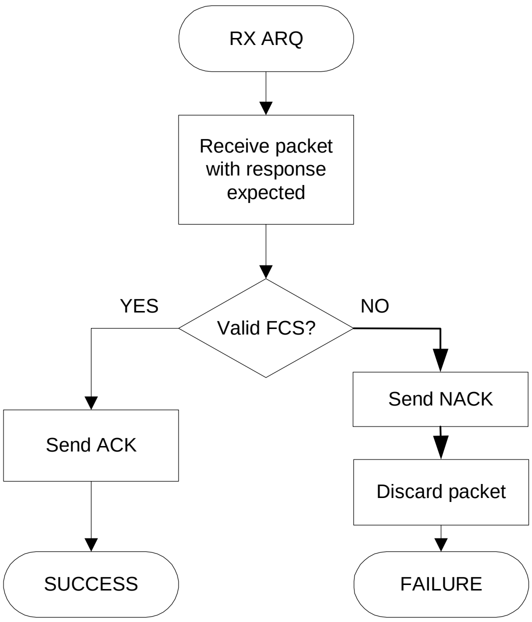
All nodes will detect ACK during response time but only one station expecting ACK will accept it as acknowledgement and use 16 bit of FCS from ACK for identification.

MAC acknowledgement is described in details in 7.3.2.



2288

2289 **Figure D.4 – Transmit ARQ**



2290

2291 **Figure D.5 – Receive ARQ**

D.6 Segmentation and reassembly overview

Since PHY specification supports different types of modulation and tone map a number of data bytes of PHY payload can be changed dynamically based on channel condition. This requires implementing MAC payload fragmentation on MAC sublayer.

If the MAC payload is too large to fit wholly within an MSDU, it must be partitioned into smaller segments that can each fit within an MSDU. This process of partitioning MAC frame into MSDU s is called segmentation.

The segmentation may require adding padding bytes to the last segment in order fill last PHY frame. The reverse process is called reassembly. The segmentation improves the probability of delivery over harsh channels and contributes to better latency characteristics for all stations by restricting the length of each individual transmission.

All forms of addressed delivery (unicast, multicast, and broadcast) are subject to segmentation. Acknowledgments and retransmissions occur independently for each segment.

The Segment Control fields: **SL**, **SC** and **LSF** are used to keep track of segments of fragmented packet and assembly whole packet on receiver side.

Table D.1 – Segment control fields

Field	Byte	Bit number	Bits	Definition
RES	0	7-4	4	Reserved
TMR	0	3	1	Tone map request: 1: Tone map is requested 0: Tone map is not requested
CC	0	2	1	Contention Control: 0: contention is allowed in next contention state 1: contention free access
CAP	0	1	1	Channel access priority: 0: Normal 1: High
LSF	0	0	1	Last Segment Flag is set for last segment only
SC	1	7-2	6	Segment Count
SL[9-8]	1	1-0	2	Segment Length of MAC frame
SL[7-0]	2	7-0	8	Segment Length of MAC frame

Annex E
(normative)

Modified MAC sublayer data primitives

E.1 MCPS-DATA.request

The semantic of the MCPS-DATA.request primitive is as follows:

```
MCPS-DATA.request    (
                       SrcAddrMode,
                       DstAddrMode,
                       DstPANId,
                       DstAddr,
                       msduLength,
                       msdu,
                       msduHandle,
                       TxOptions,
                       SecurityLevel,
                       KeyIdMode,
                       KeySource,
                       KeyIndex,
                       QualityOfService
                       )
```

Table E.1 specifies the parameters for the MCPS-DATA.request primitive.

Table E.1 – MCPS-DATA.request parameters

Name	Type	Valid range	Description
SrcAddrMode	Integer	0x00–0x03	The source addressing mode for this primitive and subsequent MPDU. This value can take one of the following values: 0x00 = no address (addressing fields omitted, see 7.2.1.1.8). 0x01 = reserved. 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
DstAddrMode	Integer	0x00–0x03	The destination addressing mode for this primitive and subsequent MPDU. This value can take one of the following values: 0x00 = no address (addressing fields omitted, see 7.2.1.1.6). 0x01 = reserved. 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
DstPANId	Integer	0x0000–0xffff	The 16-bit PAN identifier of the entity to which the MSDU is being transferred.

Name	Type	Valid range	Description
DstAddr	Device address	As specified by the DstAddrMode parameter	The individual device address of the entity to which the MSDU is being transferred.
msduLength	Integer	□*aMaxMACPayloadSize*	The number of octets contained in the MSDU to be transmitted by the MAC sublayer entity.
msdu	Set of octets	—	The set of octets forming the MSDU to be transmitted by the MAC sublayer entity.
msduHandle	Integer	0x00–0xff	The handle associated with the MSDU to be transmitted by the MAC sublayer entity.
TxOptions	Bitmap	3-bit field	The 3 bits (b0 , b1 , b2) indicate the transmission options for this MSDU. For b0: 1 = acknowledged transmission, 0 = unacknowledged transmission. For b1: 1 = GTS transmission, 0 = CAP transmission for a beacon-enabled PAN. For b2: 1 = indirect transmission, 0 = direct transmission. For a non beacon-enabled PAN, bit b1 should always be set to 0.
QualityOfService	Integer	0x00–0x02	The QOS (Quality of Service) parameter of the MSDU to be transmitted by the MAC sublayer entity. This value can take one of the following values: 0 = Normal priority, 1 = High priority, 2 = Contention free.
SecurityLevel	Integer	0x00–0x07	The security level to be used (see 7.3.6).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (see 7.3.6).This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8octets	As specified by the KeyIdMode parameter	The originator of the key to be used (see 7.3.6).This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.3.6). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.

E.2 MCPS-DATA.indication

The semantic of the MCPS-DATA.indication primitive is as follows:

```
MCPS-DATA.request    (
                        SrcAddrMode,
                        SrcPANId,
                        SrcAddr,
                        DstAddrMode,
```

```
DstPANId,
DstAddr,
msduLength,
msdu,
msduLinkQuality,
DSN,
Timestamp,
SecurityLevel,
KeyIdMode,
KeySource,
KeyIndex,
QualityOfService
)
```

The table below specifies the parameters for the MCPS-DATA.indication primitive.

Table E.2 – MCPS-DATA.request parameters

Name	Type	Valid range	Description
SrcAddrMode	Integer	0x00 – 0x03	The source addressing mode for this primitive and subsequent MPDU. This value can take one of the following values: 0x00 = no address (addressing fields omitted, see clause 7.2.1.1.8 of IEEE 802.15.4). 0x01 = reserved. 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
SrcPANId	Integer	0x0000 – 0xFFFF	The 16-bit PAN identifier of the device from which the frame was received.
SrcAddr	Device address	As specified by the SrcAddrMode parameter	The address of the device which sent the message.
DstAddrMode	Integer	0x00 – 0x03	The destination addressing mode for this primitive and subsequent MPDU. This value can take one of the following values: 0x00 = no address (addressing fields omitted, see clause 7.2.1.1.6 of IEEE 802.15.4). 0x01 = reserved. 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
DstPANId	Integer	0x0000–0xffff	The 16-bit PAN identifier of the entity to which the MSDU is being transferred.
DstAddr	Device address	As specified by the DstAddrMode parameter	The individual device address of the entity to which the MSDU is being transferred.
msduLength	Integer	≤ aMaxMACPayloadSize	The number of octets contained in the MSDU to be indicated to the upper layer
msdu	Set of octets	—	The set of octets forming the MSDU received by the MAC sublayer entity.
msduLinkQuality	Integer	0x00 – 0xFF	The LQI value measured during reception of the message.
DSN	Integer	0x00 – 0xFF	The DSN of the received frame.

Name	Type	Valid range	Description
Timestamp	Integer	0x00000000 – 0xFFFFFFFF	The time, in symbols, at which the frame was received.
SecurityLevel	Integer	0x00–0x07	The security level used (see Table 95 in clause 7.6.2.2.1 of IEEE 802.15.4).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key used (see Table 96 in clause 7.6.2.2.2 of IEEE 802.15.4). This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8octets	As specified by the KeyIdMode parameter	The originator of the key used (see clause 7.6.2.4.1 of IEEE 802.15.4). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key used (see clause 7.6.2.4.2 of IEEE 802.15.4). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
QualityOfService	Integer	0x00–0x02	The QOS (Quality of Service) parameter of the MSDU received by the MAC sublayer entity. This value can take one of the following values: 0 = Normal priority; 1 = High priority; 2 = Contention free;

Annex F
(normative)

MAC acknowledgement

The present specification does not use IEEE802.15.4-2006 MAC acknowledgment frame but specifies a positive and negative acknowledgments using Frame Control Header (section 5.5 of PHY specification).

The Frame Control Header contains an information used by all stations in the network for channel access, as well as PHY receiver information used by the destination. For this reason, Frame Control Header has specific physical layer encoding and modulation as defined in PHY specification.

Only Frame Control Header will be used as positive (ACK) or negative (NACK) acknowledgement.

The packet originator may request an acknowledgment by setting Delimiter Type field of Frame Control Header (section 5.5 of PHY specification).

The receiver will send ACK to the originator if it is requested and the MAC frame was decoded correctly by PHY.

The receiver will send NACK to the originator if it is requested and the received MAC frame is corrupted and cannot be recovered by PHY.

ACK and NACK frames contain the 16-bit CRC (MAC FCS field) received in the MAC frame for which the ACK or NACK response is being sent. These 16 bits are used as ACK or NACK identifier and located in 2 bytes of FCH (TM[0:7] and PDC) (see 5.5 of PHY specification). The transmitter will compare against transmitted FCS to determine validity of the response. If it matches of transmitted FCS, the response is accepted. If it does not match the FCS, the response is ignored and treated as a collision.

Annex G
(normative)

Adaptation sublayer service primitives

G.1 ADP Data service

G.1.1 Overview

The ADPD is used to transport application layer PDU to other devices on the network, and supports the following primitives:

– ADPD-DATA.request;

– ADPD-DATA.confirm;

– ADPD-DATA.indication.

G.1.2 ADPD-DATA.request

G.1.2.1 Semantics of the service primitive

This primitive requests the transfer of an application PDU to another device, or multiple devices. The semantics of this primitive are as follows:

```
ADPD-DATA.request    (
                     NsduLength,
                     Nsdu,
                     NsduHandle,
                     DiscoverRoute,
                     QualityOfService,
                     SecurityEnabled
                     )
```

Table G.1 – Parameters of the ADPD-DATA.request primitive

Name	Type	Valid Range	Description
NsduLength	Integer	0 – 1 280	The size of the NSDU, in bytes
Nsdu	Set of octets	-	The NSDU to send
NsduHandle	Integer	0x00 - 0xFF	The handle of the NSDU to transmit. This parameter is used to identify in the ADPD-DATA.confirm primitive which request is concerned. It can be randomly chosen by the application layer.
DiscoverRoute	Boolean	TRUE or FALSE	If TRUE, a route discovery procedure will be performed prior to sending the frame if a route to the destination is not available in the routing table. If FALSE, no route discovery is performed.
QualityOfService	Integer	0x00 - 0x01	The required quality of service (QoS) of the frame to send. Allowed values are: 0x00 = standard priority 0x01 = high priority
SecurityEnabled	Boolean	TRUE or FALSE	If TRUE, this parameter enables the adaptation sublayer security for processing the frame.

G.1.2.2 When generated

This primitive is generated by the upper layer to request the sending of a given NSDU.

G.1.2.3 Effect on receipt

If this primitive is received when the device has not joined a network, the adaptation sublayer will issue an ADPD-DATA.confirm primitive with the status INVALID_REQUEST. Else, the ADPD constructs a 6LoWPAN frame with the following characteristics depending on transmission mode:

- Case of unicast frame:
  - The mesh addressing header is present as described in clause 5.2 of RFC 4944, where:
    - V must be set to 1, to specify that the originator address is a 16-bit network address;
    - F must be set to 1, to specify that the originator address is a 16-bit network address;
    - HopsLft = MaxHops;
    - Originator address = The 16-bit network address of the sending device, available in the NIB;
    - Final destination address = 16-bit destination address of the device designated by the IPv6 address “DstAddr”.
  - The broadcast header is not present,
  - If necessary, the fragmentation header must be present to transport NPDU which do not fit in an entire IEEE 802.15.4 frame. In that case, clause 5.3 of RFC 4944 applies,
  - LOWPAN_HC1 compressed IPv6 header is present with the following parameters:
    - IPv6 Source Address mode = PC-IC (bits 0 and 1 set to 1);
    - IPv6 Destination Address mode = PC-IC (bits 2 and 3 set to 1);
    - Bit 4 = 1 (no Traffic Class and Flow Label);
    - Bits 5 and 6 = value of NsduType.
- Case of multicast frame:
  - The mesh addressing header is present as described in part 5.2 of [RFC4944], where
    - V must be set to 1, to specify that the originator address is a 16-bit network address;
    - F must be set to 1, to specify that the originator address is a 16-bit network address;
    - HopsLft = MaxHops;
    - Originator address = The 16-bit network address of the sending device, available in the NIB;
    - Final destination address = 0xFFFF;
    - The broadcast header is present with the following values:
      - Sequence Number = previous Sequence Number + 1
  - If necessary, the fragmentation header must be present to transport NPDU which do not fit in an entire IEEE 802.15.4 frame. In that case, clause 5.3 of RFC 4944 applies,
  - LOWPAN_HC1 compressed IPv6 header is present with the following parameters:
    - IPv6 Source Address mode = PC-IC (bits 0 and 1 set to 1);
    - IPv6 Destination Address mode = PC-IC (bits 2 and 3 set to 1);
    - Bit 4 = 1 (no Traffic Class and Flow Label);

– Bits 5 and 6 = value of NsduType.

Once the frame is constructed, it is routed according to the procedures described in 7.4.4 (modified clauses 6 of draft-daniel-6lowpan-load-adhoc-routing-03) if the destination address is a unicast address. If the frame is to be transmitted, the MCPS-Data.request primitive is invoked, with the following parameters in case of a unicast sending:

– SrcAddrMode = 0x02, for 16-bit address;
– DstAddrMode = 0x02, for 16-bit address;
– SrcPANId = DstPANId = the value of macPANId obtained from the MAC PIB;
– SrcAddr = the value of macShortAddr obtained from the MAC PIB;
– DstAddr = the 16-bit address of the next hop determined by the routing procedure;
– msduLength = the length of the frame, or fragment in case of fragmentation, in bytes;
– msdu = the frame itself;
– msduHandle = NsduHandle;
– TxOptions:
  - b0 = 1 if unicast transmission, 0 otherwise;
  - b1 = 0;
  - b2 = 0.
– SecurityLevel:
  - 0 if SecurityEnabled = FALSE;
  - 5 if SecurityEnabled = TRUE;
– KeyIdMode, KeySource : Ignored,
– KeyIndex : Ignored if SecurityLevel=0; Else depends on security policy.

In case of a broadcast (or multicast) frame, the MCPS-Data.request primitive is invoked with the following parameters:

– SrcAddrMode = 0x02, for 16-bit address;
– DstAddrMode = 0x02, for 16-bit address;
– SrcPANId = DstPANId = the value of macPANId obtained from the MAC PIB;
– SrcAddr = the value of macShortAddr obtained from the MAC PIB;
– DstAddr = 0xFFFF;
– msduLength = the length of the frame, or fragment in case of fragmentation, in bytes;
– msdu = the frame itself;
– msduHandle = NsduHandle;
– TxOptions:
  - b0 = 1 if unicast transmission, 0 otherwise,
  - b1 = 0,
  - b2 = 0.
– SecurityLevel
  - 0 if SecurityEnabled = FALSE,
  - 5 if SecurityEnabled = TRUE.
– KeyIdMode, KeySource : Ignored;
– KeyIndex : Ignored if SecurityLevel=0; Else depends on security policy.

If security processing fails for that frame, it must be discarded and an ADPD-DATA.confirm primitive must be generated with the status code returned by the security processing suite.

If the DiscoverRoute parameter is set to TRUE then, the route discovery procedure should be initiated prior to sending the frame in case the final destination address is not available in the routing table. For a complete description of this procedure, see 7.4.4.

G.1.3 ADPD-DATA.confirm

G.1.3.1 Semantics of the service primitive

This primitive reports the result of a previous ADPD-DATA.request primitive.

The semantics of this primitive are as follows:

```
ADPD-DATA.confirm    (
                     Status,
                     NsduHandle
                     )
```

Table G.2 – Parameters of the ADPD-DATA.confirm primitive

Name	Type	Valid Range	Description
Status	Enum	SUCCESS, INVALID_IPV6_FRAME, INVALID_REQUEST, NO_KEY, BAD_CCM_OUTPUT, ROUTE_ERROR, BT_TABLE_FULL, FRAME_NOT_BUFFERED or any status values returned from security suite or the MCPS-DATA.confirm primitive	The status code of a previous ADPD-DATA.request identified by its NsduHandle
NsduHandle	Integer	0x00 - 0xFF	The handle of the NSDU confirmed by this primitive

G.1.3.2 When generated

This primitive is generated in response to an ADPD-DATA.request primitive. The Status parameter indicates if the request succeeded or the reason of failure.

G.1.3.3 Effect on receipt

On reception of this primitive, the upper layer is notified of the status of a previous ADPD-DATA.request primitive.

G.1.4 ADPD-DATA.indication

G.1.4.1 Semantics of the service primitive

This primitive is used to transfer received data from the adaptation sublayer to the upper layer. The semantics of this primitive are as follows:

```
ADPD-DATA.indication    (
                     NsduLength,
```

Nsdu,
LinkQualityIndicator,
SecurityEnabled
)

Table G.3 – Parameters of the ADPD-DATA.indication primitive

Name	Type	Valid Range	Description
NsduLength	Integer	0-1280	The size of the NSDU, in bytes
Nsdu	Set of octets	-	The received NSDU
LinkQualityIndicator	Integer	0x00-0xFF	The value of the link quality during reception of the frame
SecurityEnabled	Boolean	TRUE or FALSE	TRUE if the frame was received using security.

G.1.4.2 When generated

This primitive is generated by the adaptation sublayer when a valid data frame whose final destination is the current station has been received.

G.1.4.3 Effect on receipt

On generation of this primitive, the upper layer is notified of the arrival of a data frame.

G.2 ADP Management service

G.2.1 Overview

The ADPM allows the transport of command frames used for network maintenance. The list of primitives supported by the ADPM is:

- ADPM-DISCOVERY.request;
- ADPM-DISCOVERY.confirm;
- ADPM-NETWORK-START.request;
- ADPM-NETWORK-START.confirm;
- ADPM-NETWORK-JOIN.request;
- ADPM-NETWORK-JOIN.confirm;
- ADPM-NETWORK-JOIN.indication;
- ADPM-NETWORK-LEAVE.request;
- ADPM-NETWORK-LEAVE.indication;
- ADPM-NETWORK-LEAVE.confirm;
- ADPM-RESET.request;
- ADPM-RESET.confirm;
- ADPM-GET.request;
- ADPM-GET.confirm;
- ADPM-SET.request;
- ADPM-SET.confirm;
- ADPM-NETWORK-STATUS.indication;
- ADPM-ROUTE-DISCOVERY.request;
- ADPM-ROUTE-DISCOVERY.confirm.

G.2.2 ADPM-DISCOVERY.request

G.2.2.1 Semantics of the service primitive

This primitive allows the upper layer to request the ADPM to scan for networks operating in its POS.

The semantics of this primitive are as follows:

```
ADPM-DISCOVERY.request   (
                                Duration,
                                )
```

Table G.4 – Parameters of the ADPM-DISCOVERY.request primitive

Name	Type	Valid Range	Description
Duration	Integer	0x00-0xFF	The number of seconds the active scan must last

G.2.2.2 When generated

This primitive is generated by the next upper layer to get informed of the current networks operating in the POS of the device.

G.2.2.3 Effect on receipt

On receipt of this primitive, the ADP layer will initiate an active scan by invoking the MLME-SCAN.request with the following parameters:

- ScanType = 0x01 for active scan;
- ScanChannels = all bits set to 0 (not used);
- ScanDuration = Duration;
- ChannelPage = 0 (not used);
- SecurityLevel = 0;
- KeyIdMode, KeySource and KeyIndex: Ignored.

On receipt of the MLME-SCAN.confirm primitive, the ADP layer issues an ADPM-DISCOVERY.confirm primitive containing the PAN ID of all the PANs operating in the POS of the device, or an error code.

G.2.3 ADPM-DISCOVERY.confirm

G.2.3.1 Semantics of the service primitive

This primitive is generated by the ADP layer upon completion of a previous ADPM-DISCOVERY.request.

The semantics of this primitive are as follows:

```
ADPM-DISCOVERY.confirm   (
                                Status,
                                PANCount,
                                PANDescriptor
                                )
```

Table G.5 – Parameters of the ADPM-DISCOVERY.confirm primitive

Name	Type	Valid Range	Description

Name	Type	Valid Range	Description
Status	Enum	Any status returned by the MLME-SCAN.confirm primitive	See IEEE 802.15.4 for the complete list of status codes and their meaning
PANCount	Integer	0x00-0xFF	The number of networks operating in the POS of the device
PANDescriptior	List of PAN descriptors	This list contains the PAN descriptors as described in Table .	The PAN operating in the POS of the device

Table G.6 – PAN descriptor structure specification

Name	Type	Valid Range	Description
ExtendedPANId	List of integers	This list contains the PAN IDs of the network found. The size of the list is PANCount. Each ExtendedPANId must be in the range 0x000000000001 – 0xFFFFFFFFFFFE	The list of 64-bit PAN identifiers.
PANId	List of integers	This list contains the 16-bit PAN IDs of the network found. The size of the list is PANCount, and its elements appear in the same order as the ExtendedPANId list. Each PANId must be in the range 0x0000 – 0xFFFF	The list of 16-bit PAN identifiers.

G.2.3.2 When generated

This primitive is generated by the ADP layer for the upper layer on completion of an ADPM-DISCOVERY.request primitive.

G.2.3.3 Effect on receipt

On receipt of this primitive, the upper layer is notified of the completion of the network scan, and obtains a list of found operating networks.

G.2.4 ADPM-NETWORK-START.request

G.2.4.1 Semantics of the service primitive

This primitive allows the upper layer to request the starting of a new network. It must only be invoked by device designated as the PAN coordinator during the factory process.

The semantics of this primitive are as follows:

```
ADPM-NETWORK-START.request (
                              PANId
                              )
```

Table G.7 – Parameters of the ADPM-NETWORK-START.request primitive

Name	Type	Valid Range	Description
PANId	Integer	0x0000 – 0xFFFF	The PANId of the network to create, determined at the application level

G.2.4.2 When generated

This primitive is generated by the upper layer of the PAN coordinator to start a new network.

G.2.4.3 Effect on receipt

On receipt of this primitive by a device which is not a PAN coordinator, it must issue an ADPM-NETWORK-START.confirm primitive with the status INVALID_REQUEST.

Prior to invoking this primitive, the upper layer of the PAN coordinator should perform an ADPM-DISCOVERY.request to make sure no other network is currently operating. In case another network is operating, the upper layer may invoke the ADPM-NETWORK-START.request.

On receipt of this primitive by a device which is the PAN coordinator, and if no network has already be formed, the ADP layer must perform the steps described in 7.5.1.

On receipt of the MLME-START.confirm primitive, the ADP layer must issue an ADPM-NETWORK-START.confirm primitive with the appropriate status code.

G.2.5 ADPM-NETWORK-START.confirm

G.2.5.1 Semantics of the service primitive

This primitive reports the status of a ADPM-NETWORK-START.request.

The semantics of this primitive are as follows:

```
ADPM-NETWORK-START.confirm (
                            Status
                            )
```

Table G.8 – Parameters of the ADPM-NETWORK-START.confirm primitive

Name	Type	Valid Range	Description
Status	Enum	SUCCESS, INVALID_REQUEST, STARTUP_FAILURE or any status value returned from the MLME-START.confirm primitive	The result of the attempt to create the network

G.2.5.2 When generated

This primitive is generated by the ADP layer in response to an ADPM-NETWORK-START.request primitive, and indicates if the network formation was successful or not, and an eventual reason for failure.

G.2.5.3 Effect on receipt

On receipt of this primitive, the next higher layer is notified about the status of its previous ADPM-NETWORK-START.request.

G.2.6 ADPM-NETWORK-JOIN.request

G.2.6.1 Semantics of the service primitive

This primitive allows the next upper layer to join an existing network.

The semantics of this primitive are as follows:

```
ADPM-NETWORK-JOIN.request    (
                       PANId,
                       LBAAddress
                       )
```

Table G.9 – Parameters of the ADPM-NETWORK-JOIN.request primitive

Name	Type	Valid Range	Description
PANId	Integer	0x0000 – 0xFFFF	The 16-bit PAN identifier of the network to join.
LBAAddress	16-bit address	0x0000 – 0xFFFF	The 16-bit short address of the device acting as a LoWPAN Bootstrap Agent as defined in draft-6lowpan-commissioning-02.

G.2.6.2 When generated

The upper layer invokes this primitive when it wishes to join an existing PAN using the MAC association procedure.

G.2.6.3 Effect on receipt

On receipt of this primitive by a device which is already joined, the adaptation sublayer generates an ADPM-NETWORK-JOIN.confirm with the status INVALID_REQUEST.

On receipt of this primitive by a device which is not already joined, the adaptation sublayer initiates the MAC association procedure (“bootstrap”) described in 7.4.5.2.2.

On completion, an MLME-SET.request is invoked to set the 16-bit short address of the device which was obtained during the “bootstrapping” phase. Then, an ADPM-NETWORK-JOIN.confirm primitive is generated with a status of SUCCESS.

G.2.7 ADPM-NETWORK-JOIN.confirm

G.2.7.1 Semantics of the service primitive

This primitive is generated by the ADP layer to indicate the completion status of a previous ADPM-NETWORK-JOIN.request.

The semantics of this primitive are as follows:

```
ADPM-NETWORK-JOIN.confirm   (
                            Status,
                            NetworkAddress,
                            PANId
                            )
```

Table G.10 – Parameters of the ADPM-NETWORK-JOIN.confirm primitive

Name	Type	Valid Range	Description
Status	Status	SUCCESS, INVALID_REQUEST, NOT_PERMITTED.	The result of the attempt to join the network.
NetworkAddress	Integer	0x0001 – 0xFFF7 and 0xFFFF	The 16-bit network address that was allocated to the device. If the allocation fails, this address is equal to 0xFFFF.
PANId	Integer	0x0000 – 0xFFFF	The 16-bit address of the PAN of which the device is now a member.

G.2.7.2 When generated

This primitive is generated in response to an ADPM-NETWORK-JOIN.request primitive, and allows the upper layer to obtain information on the status of its request.

The status NOT_PERMITTED is given if the device was unable to authenticate itself to the PAN coordinator.

G.2.7.3 Effect on receipt

On receipt of this primitive the upper layer is informed on the status of its request.

G.2.8 ADPM-NETWORK-JOIN.indication

G.2.8.1 Semantics of the service primitive

This primitive allows the upper layer of the PAN coordinator to be notified when a new device has successfully completed the association procedure, and is now part of the network.

The semantics of this primitive are as follows:

```
ADPM-NETWORK-JOIN.indication (
                              NetworkAddress,
                              ExtendedAddress,
                              CapabilityInformation,
                              )
```

Table G.11 – Parameters of the ADPM-NETWORK-JOIN.indication primitive

Name	Type	Valid Range	Description
NetworkAddress	IPv6 address	See RFC 4944	The IPv6 network address of the device that was added to the network. This address was given during the association procedure.
ExtendedAddress	64-bit address	0x000000000001 – 0xFFFFFFFFFFFE	The 64-bit address of the device that was added to the network. This address is unique for any device.
CapabilityInformation	Bitmap	See Figure 56 of IEEE 802.15.4	The capability information field of the device.

G.2.8.2 When generated

G.2.8.3 Effect on receipt

This primitive is generated by the ADP layer upon successful completion of the association of a new device.

The upper layer is notified of the completion of the association.

G.2.9 ADPM-NETWORK-LEAVE.request

This primitive allows the PAN coordinator to remove a device from the network, or allows a device to remove itself from the network.

G.2.9.1 Semantics of the service primitive

The semantics of this primitive are as follows:

```
ADPM-NETWORK-LEAVE.request (
                            ExtendedAddress
                            )
```

Table G.12 – Parameters of the ADPM-NETWORK-LEAVE.request primitive

Name	Type	Valid Range	Description
ExtendedAddress	64-bit address	Any	The 64-bit network address of the device to remove from the network. If NULL, the device removes itself from the network.

G.2.9.2 When generated

The next higher layer generates this primitive to leave the network, or to request another device to do so.

G.2.9.3 Effect on receipt

On receipt of this primitive by a device which is not associated to any network, the adaptation sublayer must issue an ADPM-NETWORK-LEAVE.confirm primitive with the status INVALID_REQUEST.

On receipt of this primitive by a device which is associated to any network, the following steps must be performed:

- If the device is a coordinator
  - If ExtendedAddress == NULL
    - Issue ADPM-NETWORK-LEAVE.confirm with INVALID_REQUEST
  - Else
    - If the device exists
      - Remove the device which has the address ExtendedAddress from the network using the procedure described in 7.4.5.2.2.7.
      - Issue ADPM-NETWORK-LEAVE.confirm with SUCCESS
    - If the device doesn’t exist
      - Issue ADPM-NETWORK-LEAVE.confirm with UNKNOWN_DEVICE
- Else (device is not a coordinator)
  - If ExtendedAddress == NULL
    - The device removes itself from the network, using the procedure described in 7.4.5.2.2.8.
    - Issue ADPM-NETWORK-LEAVE.confirm with SUCCESS
  - Else
    - Issue ADPM-NETWORK-LEAVE.confirm with INVALID_REQUEST

G.2.10 ADPM-NETWORK-LEAVE.indication

G.2.10.1 Semantics of the service primitive

This primitive is generated by the ADP layer to inform the upper layer that a device has been unregistered from the network.

The semantics of this primitive are as follows:

```
ADPM-NETWORK-LEAVE.indication (
                    ExtendedAddress,
                    )
```

Table G.13 – Parameters of the ADPM-NETWORK-LEAVE.indication primitive

Name	Type	Valid Range	Description

Name	Type	Valid Range	Description
ExtendedAddress	64-bit address	Any	The 64-bit network address of the device removed from the network.

G.2.10.2 When generated

This primitive is generated by the adaptation sublayer of a device when it has been removed from the network by the PAN coordinator or by the adaptation sublayer of the PAN coordinator when a device has decided to leave the network.

G.2.10.3 Effect on receipt

On receipt of this primitive, the upper layer of the device is notified that it is no more a part of the PAN.

G.2.11 ADPM-NETWORK-LEAVE.confirm

G.2.11.1 Semantics of the service primitive

This primitive allows the upper layer to be informed on the status of its previous ADPM-NETWORK-LEAVE.request. This request can be either to leave by itself the network, or to instruct another device to leave (PAN coordinator only).

The semantics of this primitive are as follows:

```
ADPM-NETWORK-LEAVE.confirm  (
                                Status,
                                ExtendedAddress
                                )
```

Table G.14 – Parameters of the ADPM-NETWORK-LEAVE.confirm primitive

Name	Type	Valid Range	Description
Status	Enum	SUCCESS, INVALID_REQUEST, UNKNOWN_DEVICE or any status returned by the MCPS-DATA.confirm primitive	The status of the request.
ExtendedAddress	64-bit address	Any	The 64-bit network address of the device removed from the network.

G.2.11.2 When generated

This primitive is generated on completion of a device removal. If it is successful, the SUCCESS code is given. Else, an error status is given as explained in 7.4.5.2.2.8.

G.2.11.3 Effect on receipt

On receipt, the upper layer is notified of the result of its request.

G.2.12 ADPM-RESET.request

G.2.12.1 Semantics of the service primitive

This primitive allows the upper layer to request that the ADP layer performs a reset.

The semantics of this primitive are as follows:

```
ADPM-RESET.request  (
                    )
```

This primitive has no parameter.

G.2.12.2 When generated

This primitive allows a reset of the adaptation sublayer, and allows resetting the NIB attributes.

G.2.12.3 Effect on receipt

On receipt of this primitive, the following steps are performed:

- The adaptation sublayer issues a MLME-RESET.request primitive with the SetDefaultPIB parameter set to TRUE, and waits for the MLME-RESET.confirm primitive;
- The adaptation sublayer clears all of its internal variables, and flushes its routing and neighbor tables;
- The adaptation sublayer issues an ADPM-RESET.confirm primitive with the status SUCCESS, or DISABLE_TRX_FAILURE if the MAC reset operation failed.

G.2.13 ADPM-RESET.confirm

G.2.13.1 Semantics of the service primitive

This primitive allows the upper layer to be notified of the completion of an ADPM-RESET.request primitive.

The semantics of this primitive are as follows:

```
ADPM-RESET.confirm (
                Status
                )
```

Table G.15 – Parameters of the ADPM-RESET.confirm primitive

Name	Type	Valid Range	Description
Status	Enum	Any status value returned from the MLMERESET.confirm primitive	The status of the request

G.2.13.2 When generated

This primitive is generated by the ADP layer when a previous ADPM-RESET.request primitive has completed.

G.2.13.3 Effect on receipt

The upper layer is notified of the completion of the command.

G.2.14 ADPM-GET.request

G.2.14.1 Semantics of the service primitive

This primitive allows the upper layer to get the value of an attribute from the information base.

The semantics of this primitive are as follows:

```
ADPM-GET.request (
                AttributeId,
                AttributeIndex
                )
```

Table G.16 – Parameters of the ADPM-GET.request primitive

Name	Type	Valid Range	Description
AttributeId	Integer	See clause 7.4.2	The identifier of the IB attribute to read
AttributeIndex	Integer	Depends on attribute, see 7.4.2	The index within the table of the specified IB attribute to read. This parameter is valid only for IB attributes that are tables.

G.2.14.2 When generated

This primitive is generated by the upper layer to read the value of an attribute from the IB.

G.2.14.3 Effect on receipt

On receipt of this primitive, the adaptation sublayer attempts to retrieve the selected attribute in the information base. If the attribute is not found, the adaptation layer generates an ADPM-GET.confirm primitive with the status UNSUPPORTED_ATTRIBUTE. If the attribute is found (and is a table), but the AttributeIndex is out of range, the adaptation layer generates an ADPM-GET.confirm primitive with the status INVALID_INDEX.

Else, the adaptation sublayer generates an ADPM-GET.confirm primitive with the status SUCCESS, and the value read from the IB in the AttributeValue parameter.

G.2.15 ADPM-GET.confirm

G.2.15.1 Semantics of the service primitive

This primitive allows the upper layer to be informed of the status of a previously issued ADPM-GET.request primitive.

The semantics of this primitive are as follows:

ADPM-GET.confirm (
    Status,
    AttributeId,
    AttributeIndex,
    AttributeValue
)

Table G.17 – Parameters of the ADPM-GET.confirm primitive

Name	Type	Valid Range	Description
Status	Enum	SUCCESS, UNSUPPORTED_ATTRIBUTE or INVALID_INDEX	The status of the reading.
AttributeId	Integer	See 7.4.2	The identifier of the IB attribute read.
AttributeIndex	Integer	Depends on attribute, see 7.4.2	The index within the table of the specified IB attribute read. This parameter is valid only for IB attributes that are tables.
AttributeValue	Various	Attribute specific	The value of the attribute read from the IB.

G.2.15.2 When generated

This primitive is generated by the adaptation sublayer in response to an ADPM-GET.request primitive.

G.2.15.3 Effect on receipt

On reception of this primitive, the upper layer is informed on the status of its request, and eventually gets the desired value.

G.2.16 ADPM-SET.request

G.2.16.1 Semantics of the service primitive

This primitive allows the upper layer to set the value of an attribute in the information base.

The semantics of this primitive are as follows:

```
ADPM-SET.request (
            AttributeId,
            AttributeIndex,
            AttributeValue
            )
```

Table G.18 – Parameters of the ADPM-SET.request primitive

Name	Type	Valid Range	Description
AttributeId	Integer	See 7.4.2	The identifier of the IB attribute to write
AttributeIndex	Integer	Depends on attribute, see 7.4.2	The index within the table of the specified IB attribute to write. This parameter is valid only for IB attributes that are tables.
AttributeValue	Various	Depends on attribute	The value to write

G.2.16.2 When generated

This primitive is generated by the upper layer to write the value of an attribute in the IB.

G.2.16.3 Effect on receipt

On receipt of this primitive, the adaptation sublayer attempts to retrieve the selected attribute in the information base. If the attribute is not found, the adaptation layer generates an ADPM-SET.confirm primitive with the status UNSUPPORTED_ATTRIBUTE. If the attribute is found (and is a table), but the AttributeIndex is out of range, the adaptation layer generates an ADPM-SET.confirm primitive with the status INVALID_INDEX. If the attribute is found but is read only, the adaptation layer generates an ADPM-SET.confirm primitive with the status READ_ONLY. If the attribute is found, can be written, but the AttributeValue is out of range, the adaptation layer generates an ADPM-SET.confirm primitive with the status INVALID_PARAMETER. Else, the adaptation layer generates an ADPM-SET.confirm primitive with the status SUCCESS.

G.2.17 ADPM-SET.confirm

G.2.17.1 Semantics of the service primitive

This primitive allows the upper layer to be informed about a previous ADPM-SET.request primitive.

The semantics of this primitive are as follows:

```
ADPM-SET.confirm (
            Status,
            AttributeId,
            AttributeIndex
```

)

Table G.19 – Parameters of the ADPM-SET.confirm primitive

Name	Type	Valid Range	Description
Status	Enum	SUCCESS, UNSUPPORTED_ATTRIBUTE, READ_ONLY, INVALID_PARAMETER or INVALID_INDEX	The status of the writing
AttributeId	Integer	See 7.4.2	The identifier of the IB attribute written
AttributeIndex	Integer	Depends on attribute, see 7.4.2	The index within the table of the specified IB attribute written. This parameter is valid only for IB attributes that are tables.

G.2.17.2 When generated

This primitive is generated by the adaptation layer in response to an ADPM-SET.request primitive.

G.2.17.3 Effect on receipt

On reception of this primitive, the upper layer is informed on the status of its request.

G.2.18 ADPM-NETWORK-STATUS.indication

G.2.18.1 Semantics of the service primitive

This primitive allows the next higher layer of a PAN coordinator or a coordinator to be notified when a particular event occurs on the PAN.

The semantics of this primitive are as follows:

```
ADPM-NETWORK-STATUS.indication   (
                                  Status,
                                  AdditionalInformation
                                  )
```

Table G.20 – Parameters of the ADPM-NETWORK-STATUS.indication primitive

Name	Type	Valid Range	Description
Status	Enum	PAN_ID_CONFLICT, or any status code returned by MLME-COMM-STATUS.indication	The status or event to notify.
AdditionalInformation	String	Any string	The eventual additional information to the status or event.

G.2.18.2 When generated

This primitive is generated when the adaptation sublayer of a PAN coordinator has received a LBP message from a device on the network indicating that a PAN Id conflict is occurring. See 7.5.2 for complete description of the PAN ID conflict handling mechanism.

In that case, this primitive is never generated by the adaptation sublayer of a device which is not a PAN coordinator.

This primitive is also generated if the underlying MAC layer (of a PAN coordinator or a coordinator) generates a MLME-COMM-STATUS.indication.

G.2.18.3 Effect on receipt

On reception, the upper layer of a PAN coordinator is informed that a PAN Id conflict was detected, or that a MAC event occurred.

G.2.19 ADPM-ROUTE-DISCOVERY.request

G.2.19.1 Semantics of the service primitive

This primitive allows the upper layer to initiate a route discovery.

The semantics of this primitive are as follows:

ADPM-ROUTE-DISCOVERY.request (
DstAddr,
MaxHops
)

Table G.21 – Parameters of the ADPM-ROUTE-DISCOVERY.request primitive

Name	Type	Valid Range	Description
DstAddr	Short address	See RFC 4944	The Short unicast destination address of the route discovery.
MaxHops	Integer	0x00 – 0x07	This parameter indicates the maximum number of hops allowed for the route discovery.

G.2.19.2 When generated

This primitive is generated by the upper layer of a device to obtain a route to another device.

G.2.19.3 Effect on receipt

An ADPM-ROUTE-DISCOVERY.confirm with the status INVALID_REQUEST is generated if the DstAddr is not a unicast IPv6 address, or if the MaxHops value is out of range.

On receipt of this primitive, the device will initiate a route discovery procedure as described in 7.4.4.2.3.

G.2.20 ADPM-ROUTE-DISCOVERY.confirm

G.2.20.1 Semantics of the service primitive

This primitive allows the upper layer to be informed of the completion of a route discovery.

The semantics of this primitive are as follows:

ADPM-ROUTE-DISCOVERY.request (
Status
)

Table G.22 – Parameters of the ADPM-ROUTE-DISCOVERY.confirm primitive

Name	Type	Valid Range	Description
Status	Status	SUCCESS, INVALID_REQUEST,	The status of the route discovery.

Name	Type	Valid Range	Description
		ROUTE_ERROR	

G.2.20.2 When generated

This primitive is generated by the adaptation layer on completion of a route discovery as described in 7.4.4.2.3, and in draft-daniel-6lowpan-load-adhoc-routing-03.

G.2.20.3 Effect on receipt

On reception of this primitive, the upper layer is informed on the completion of the route discovery. If the Status value is SUCCESS, the routing table has been correctly updated with a brand new route to the desired destination, and the device may begin sending frames to that destination.

G.2.21 ADPM-PATH-DISCOVERY.request

G.2.21.1 Semantics of the service primitive

This primitive allows the upper layer to initiate a path discovery.

The semantics of this primitive are as follows:

```
ADPM-PATH-DISCOVERY.request (
                              DstAddr
                              )
```

Table G.23 – Parameters of the ADPM-PATH-DISCOVERY.request primitive

Name	Type	Valid Range	Description
DstAddr	short address	0 – 1 199	The short unicast destination address of the path discovery.

G.2.21.2 When generated

This primitive is generated by the upper layer of a device to obtain the path to another device.

G.2.21.3 Effect on receipt

An ADPM-PATH-DISCOVERY.confirm with the status INVALID_REQUEST is generated if the DstAddr is not in the routing table or after the failure of the procedure.

On receipt of this primitive, the device will initiate a path discovery procedure as described in 7.4.4.2.4.

G.2.22 ADPM-PATH-DISCOVERY.confirm

G.2.22.1 Semantics of the service primitive

This primitive allows the upper layer to be informed of the completion of a path discovery.

The semantics of this primitive are as follows:

```
ADPM-ROUTE-DISCOVERY.request      (
                              DstAddr,
                              NSDU
                              )
```

Table G.24 – Parameters of the ADPM-PATH-DISCOVERY.confirm primitive

Name	Type	Valid Range	Description
DstAddr	Short address	0 – 1 199	The Short unicast destination address of the path discovery.
NsduId	integer	N.C	The buffer containing addresses of nodes constituting the path.

G.2.22.2 When generated

This primitive is generated by the adaptation layer on completion of a path discovery as described in clause 5.4.4 of the present document.

G.2.22.3 Effect on receipt

On reception of this primitive, the upper layer is informed on the completion of the path discovery.

G.2.23 ADPM-LBP.request

G.2.23.1 Semantics of the service primitive

This primitive allows the upper layer of client to send the LBP message to server modem .

The semantics of this primitive are as follows:

```
ADPM-LBP.request      (
                       DstAddrType,
                       DstAddr,
                       NsduLength,
                       Nsdu,
                       NsduType,
                       MaxHops,
                       DiscoveryRoute,
                       QualityOfService,
                       SecurityEnable
                       )
```

Table G.25 – Parameters of the ADPM-LBP.request primitive

Name	Type	Valid Range	Description
DstAddrType	Integer	0x01 – 0x02	The type of destination address contained in the DstAddr parameter. The allowed values are: 0x01 = 2 Bytes address (LBA address) 0x02 = 8 Bytes address (LBD address
DstAddr	Set of octets	-	16 bits address of LBA or 64 bits (extended address of LBD)
NsduLength	Integer	0 – 1 280	The size of the NSDU, in bytes
Nsdu	Set of octets	-	The NSDU to send
NsduHandle	Integer	0x00 – 0xFF	The handle of the NSDU to transmit. This parameter is used to identify in the ADPM-LBP.confirm primitive which request is concerned. It can be randomly chosen by the application layer.
NsduType	Integer	0x00 – 0x03	The type of data contained in the NSDU. 0x00 = any data

Name	Type	Valid Range	Description
			0x01 = UDP 0x02 = ICMP 0x03 = TCP
MaxHops	Integer	0x00 – 0x07	The number of times the frame will be repeated by network routers.
DiscoveryRoute	Boolean	TRUE – FALSE	If TRUE, a route discovery procedure will be performed prior to sending the frame if a route to the destination is not available in the routing table. If FALSE, no route discovery is performed.
QualityOfService	Integer	0x00 – 0x01	The required quality of service (QoS) of the frame to send. Allowed values are: 0x00 = standard priority 0x01 = high priority
SecurityEnabled	Boolean	TRUE – FALSE	If TRUE, this parameter enables the ADP layer security for processing the frame.

G.2.23.2 When generated

This primitive is generated by the client LBPServer to perform the authentication, re-keying and leave procedure.

G.2.23.3 Effect on receipt

On reception of this primitive, the modem sends the coming frame to the destination.

G.2.24 ADPM-LBP.confirm

G.2.24.1 Semantics of the service primitive

This primitive reports the result of a previous ADPM-LBP.request primitive.

The semantics of this primitive are as follows:

```
ADPM-LBP.confirm      (
                       Status,
                       NsduHandle,
                     )
```

Table G.26 – Parameters of the ADPM-LBP.confirm primitive

Name	Type	Valid Range	Description
Status	Enum	SUCCESS, INVALID_REQUEST, NO_KEY, BAD_CCM_OUTPUT, ROUTE_ERROR, BT_TABLE_FULL, FRAME_NOT_BUFFERED or any status values returned from security suite or the MCPS-DATA.confirm primitive	The status code of a previous ADPM-LBP.request identified by its NsduHandle.

Name	Type	Valid Range	Description
NsduHandel	Integer	0x00 – 0xFF	The handle of the NSDU confirmed by this primitive.

G.2.24.2 When generated

This primitive is generated in response to a ADPM-LBP.request primitive, the Status parameter indicating if the request succeeded, or the reason of failure.

G.2.24.3 Effect on receipt

On reception of this primitive, the upper layer is notified of the status of a previous ADPM-LBP.request primitive.

G.2.25 ADPM-LBP.indication

G.2.25.1 Semantics of the service primitive

This primitive is used to transfer received LBP frame from the ADP layer to the upper layer.

The semantics of this primitive are as follows:

```
ADPM-LBP.request      (
                        DstAddr,
                        SrcAddr,
                        NsduLength,
                        Nsdu,
                        NsduType,
                        LinkQualityIndicator,
                        SecurityEnabled
                      )
```

Table G.27 – Parameters of the ADPM-LBP.indication primitive

Name	Type	Valid Range	Description
DstAddr	Integer	0x0000 – 0xFFFF	16 bits final destination address
SrcAddr	Integer	0x0000 – 0xFFFF	16 bits original source address
NsduLength	Integer	0 – 1 280	The size of the NSDU, in bytes
Nsdu	Set of octets	-	The NSDU to send
NsduType	Integer	0x00 – 0x03	The type of data contained in the NSDU. 0x00 = any data 0x01 = UDP 0x02 = ICMP 0x03 = TCP
LinkQualityIndicator	Integer	0x00 – 0xFF	The value of the link quality during reception of the frame
SecurityEnabled	Boolean	TRUE – FALSE	If TRUE, this parameter enables the adaptation sublayer security for processing the frame.

G.2.25.2 When generated

This primitive is generated by the ADP layer of client modem when a valid LBP frame whose final destination is the current station has been received.

G.2.25.3 Effect on receipt

On generation of this primitive, the upper layer is notified of the arrival of a LBP frame.

G.2.26 ADPM-BUFFER.indication

G.2.26.1 Semantics of the service primitive

This primitive allows the next higher layer to be notified when the modem reach his limit capability to perform coming frame.

The semantics of this primitive are as follows:

```
ADPM-BUFFER.indication    (
                           BufferReady
                           )
```

Table G.28 – Parameters of the ADPM-BUFFER.indication primitive

Name	Type	Valid Range	Description
BufferReady	Boolean	TRUE – FALSE	TRUE : modem is ready to receipt more data frame FALSE : modem is not ready, stop sending data frame

G.2.26.2 When generated

This primitive is generated when the adaptation layer of a modem has reached his limit to perform more Data frame.

G.2.26.3 Effect on receipt

On reception, the upper layer should stop the data flow if BufferReady is equal to FALSE and open it if BufferReady is TRUE.

G.3 Behavior to MAC Indications

G.3.1 Overview

This clause describes the behaviour of the adaptation layer in response to an unsolicited indication from the MAC layer.

G.3.2 MCPS-DATA.indication

On reception of this indication, the adaptation layer must execute the routing algorithm as described in 7.4.4.

G.3.3 MLME-ASSOCIATE.indication

Nothing must be done upon reception of this primitive by the adaptation layer.

G.3.4 MLME-DISASSOCIATE.indication

Nothing must be done upon reception of this primitive by the adaptation layer.

G.3.5 MLME-BEACON-NOTIFY.indication

When a MLME-BEACON-NOTIFY.indication is received, and if an ADPM-DISCOVERY.request is currently operating, the adaptation layer must add the PANId to the PANDescriptorList which will be forwarded to the upper layer in the ADPM-DISCOVERY.confirm primitive.

G.3.6 MLME-GTS.indication

Nothing must be done upon reception of this primitive by the adaptation layer.

G.3.7 MLME-ORPHAN.indication

Nothing must be done upon reception of this primitive by the adaptation layer.

G.3.8 MLME-COMM-STATUS.indication

On reception of this primitive, the adaptation layer must generate an ADPM-NETWORK-STATUS.indication primitive, with the Status parameter equal to that of the MLME-COMM-STATUS.indication primitive, and the AdditionalInformation parameter equal to the concatenation of the SrcAddr and DstAddr, separated by a “:”.

G.3.9 MLME-SYNC-LOSS.indication

Nothing must be done upon reception of this primitive by the adaptation layer.

Annex H
(normative)

Device Starting Sequence of messages

Each device should start on Not_Device_Server status and then the following procedure is performed:

a) Reset the equipment by sending the ADPM-RESET.request;
b) Set the type of the device to switch it on Device or Server mode and optionally set the PIB parameters to configure it;
   – If the equipment is a device it should perform:
c) Discovery procedure by invoking the ADPM-NETWORK-DISCOVERY.request;
d) If there is a device or a server in its pose, it must then invoke the ADPM-NETWORK-JOINNING.request to perform the bootstrapping procedure;
   – Else (the equipment is a server) it should perform:
e) Discovery procedure by invoking the ADPM-NETWORK-DISCOVERY.request.
f) If there is a device in the server's pose, it should invoke the ADPM-NETWORK-START; else it switches to joining status.

Equipment can't send or receive data or load packet unless it is joined.