International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.9903

(10/2012)

SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

Access networks – In premises networks

Narrowband orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks

Recommendation ITU-T G.9903

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TRANSMISSION MEDIA AND OPTICAL SYSTEMS CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
MULTIMEDIA QUALITY OF SERVICE AND PERFORMANCE – GENERIC AND USER-RELATED ASPECTS	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DATA OVER TRANSPORT – GENERIC ASPECTS	G.7000–G.7999
PACKET OVER TRANSPORT ASPECTS	G.8000–G.8999
ACCESS NETWORKS	G.9000–G.9999
In premises networks	**G.9900–G.9999**

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

Recommendation ITU-T G.9903

Narrowband orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks

Summary

Recommendation ITU-T G.9903 contains the physical layer (PHY) and data link layer (DLL) specification for the G3-PLC narrowband orthogonal frequency division multiplexing (OFDM) power line communication transceivers for communications via alternating current and direct current electric power lines over frequencies below 500 kHz.

This Recommendation uses material from Recommendations ITU-T G.9955 and ITU-T G.9956; specifically from Annexes A and D of ITU-T G.9955 and Annex A of ITU-T G.9956. New technical material has not been introduced in this version.

The control parameters that determine spectral content, power spectral density (PSD) mask requirements and the set of tools to support the reduction of the transmit PSD can be found in Recommendation ITU-T G.9901.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T G.9903	2012-10-29	15

FOREWORD

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

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

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

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

NOTE

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

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

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

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Page

Recommendation ITU-T G.9903

Narrowband orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks

1 Scope

Recommendation ITU-T G.9903 contains the physical layer (PHY) and data link layer (DLL) specification for the G3-PLC narrowband orthogonal frequency division multiplexing (OFDM) power line communication transceivers for communications via alternating current and direct current electric power lines over frequencies below 500 kHz. This Recommendation supports indoor and outdoor communications over low-voltage lines, medium-voltage lines, through transformer low-voltage to medium-voltage and through transformer medium-voltage to low-voltage power lines in both urban and long distance rural communications. This Recommendation addresses grid to utility meter applications, advanced metering infrastructure (AMI), and other 'Smart Grid' applications such as the charging of electric vehicles, home automation and home area networking (HAN) communications scenarios.

This Recommendation does not contain the control parameters that determine spectral content, power spectral density (PSD) mask requirements and the set of tools to support a reduction of the transmit PSD; all of which are detailed in Recommendation ITU-T G.9901.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T G.9901] Recommendation ITU-T G.9901 (2012), *Narrowband orthogonal frequency division multiplexing power line communication transceivers – Power spectral density specification*.

[IEEE 802-2001] IEEE Std 802-2001 (R2007), *IEEE Standard for Local and Metropolitan Area Networks. Overview and Architecture*.

[IEEE 802.15.4] IEEE 802.15.4:2006, *IEEE 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] IEEE 802.2:1998, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 2: Logical Link Control*.

[IETF RFC 2284] IETF RFC 2284 (1998), *PPP Extensible Authentication Protocol (EAP).*

[IETF RFC 2865] IETF RFC 2865 (2000), *Remote Authentication Dial In User Service (RADIUS).*

[IETF RFC 3748] IETF RFC 3748 (2004), *Extensible Authentication Protocol (EAP).*

[IETF RFC 4291] IETF RFC 4291 (2006), *IP Version 6 Addressing Architecture*.

[IETF RFC 4764]	IETF RFC 4764 (2007), *The EAP-PSK Protocol: A Pre-Shared Key Extensible Authentication Protocol (EAP) Method*.
[IETF RFC 4862]	IETF RFC 4862 (2007), *Ipv6 Stateless Address Autoconfiguration*.
[IETF RFC 4944]	IETF RFC 4944 (2007), *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*.

3 Definitions

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
AAA	Authentication, Authorization and Accounting
ACK	Acknowledge
ADP	Adaptation
AFE	Analogue Front End
AGC	Automatic Gain Control
AMM	Automated Meter Management
ARQ	Automatic Repeat Request
BPSK	Binary Phase Shift Keying
CC	Convolutional Code
CFA	Contention Free Access
CIFS	Contention Inter-frame Space
CP	Cyclic Prefix
CRC	Cyclic Redundancy Check
D8PSK	Differential 8 Phase Shift Keying
DBPSK	Differential Binary Phase Shift Keying
DQPSK	Differential Quadrature Phase Shift Keying
DSI	Device Specific Information
EAP	Extensible Authentication Protocol
ED	End Device
EIFS	Extended Inter-frame Space
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
HPCW	High Priority Contention Window
ICI	Inter-Carrier Interference
IFFT	Inverse Fast Fourier Transform
IFS	Inter-frame Spacing
IS	Information System
LBD	LoWPAN Bootstrapping Device
LBP	LoWPAN Boostrapping Protocol
LFSR	Linear Feedback Shift Register
LQ	Link Quality
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
MSE	Mean Square Error
NACK	Negative Acknowledgement
NIB	Neighbour Information Base
NPCW	Normal Priority Contention Window
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
PICS	Protocol Implementation Conformance Statement
PLC	Power Line Communication
PN	Pseudo Noise
POS	Personal Operating Space
PPDU	PHY Protocol Data Unit
PPM	Parts Per Million
PSDU	PHY Service Data Unit
PSI	PAN Specific Information
RADIUS	Remote Authentication Dial in User Service
RC	Repetition Code

RES	Reserved (bit fields)
RIFS	Response Inter-frame Space
rms	Root Mean Square
RS	Reed-Solomon
RX	Receiver
SC	Segment Count
S-FSK	Spread Frequency Shift Keying
SN	Sequence Number
SNR	Signal to Noise Ratio
SSCS	Service-Specific Convergence Sublayer
TMI	Tone Map Index
TMR	Tone Map Request
TX	Transmitter
TX	Transmit

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

– clause 4 of [IEEE 802.15.4]

– clause 1.2 of [IETF RFC 4944].

5 Introduction

Power line communication has been used for many decades, but a variety of new services and applications requires greater reliability and higher data rates. However, 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. Besides background noise, it is subject to impulsive noise and narrowband interference and group delays of up to several hundred microseconds.

OFDM is a modulation technique that efficiently utilizes the allowed bandwidth within the CENELEC (European Committee for Electrotechnical Standardization) band 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. OFDM-based ITU-T G.9903 specifications address the following main objectives:

1) provide robust communication in extremely harsh power line channels;

2) provide a minimum of 20 kbit/s effective data rate in the normal mode of operation;

3) ability of notching selected frequencies, allowing the cohabitation with S-FSK narrow band communication;

4) dynamic tone adoption capability to varying power line channel to ensure a robust communication.

6 General description

The following diagram illustrates an example of an AMM system.

The system provides a reliable two-way communication using an OFDM-PLC between the meters installed at the customer's premises and the concentrator, communicating in a master and slave configuration.

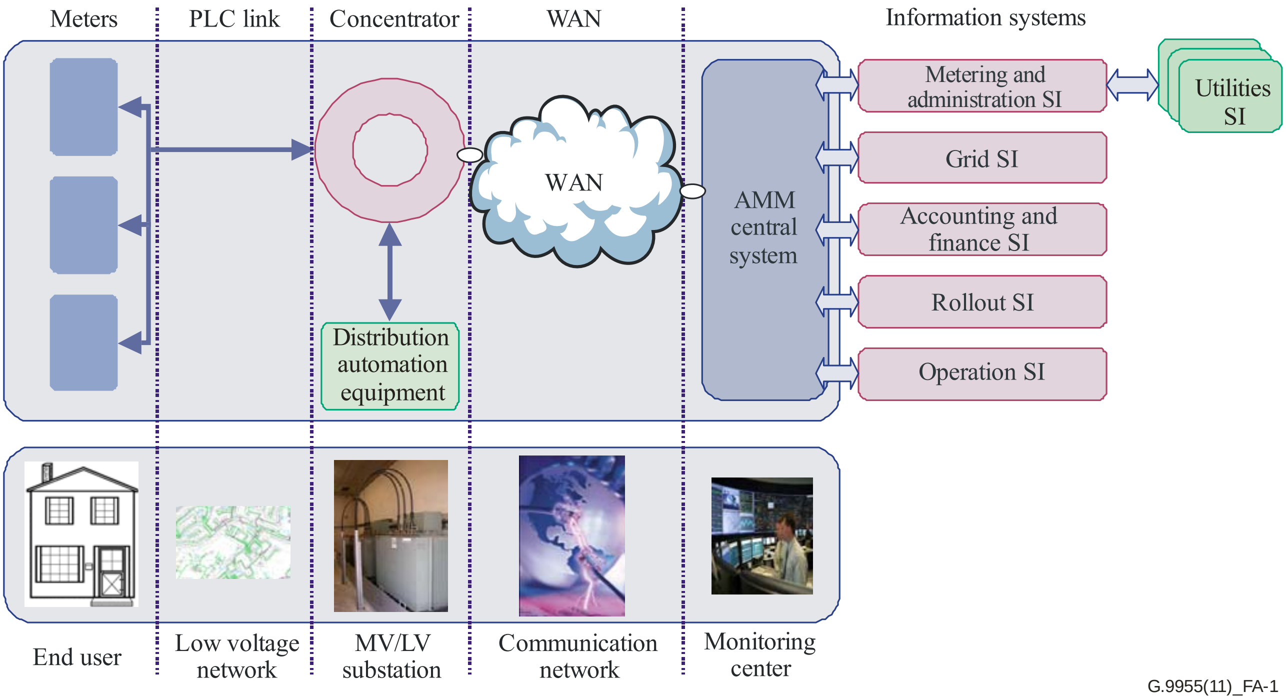


Figure 6-1 – Network architecture

The AMM architecture consists of the five following main components:

- The meter which needs to integrate the capability of measuring power consumption, simple load control and customer remote information.
- The hub which acts as an intermediary between the AMM information system and the meters. Complementary equipment supplied by the electrical network that can be connected downstream of the hub.
- The PLC (LAN) technology which allows the use of a low-voltage electrical network to exchange data and commands between meters and hubs.
- A remote connection (WAN) allows connection between the hubs and the AMM central IS.
- The central system, which not only handles its own functional services but also supplies metering services to the existing or forthcoming ENTERPRISE services (deployment IS, network IS, management-finance IS, customer-supplier IS-Intervention management IS, etc.). The customer-Supplier IS is the interface between the suppliers and AMM for handling their requirements.

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 50/60 Hz and narrowband interference and group delays of up to several hundred microseconds.

OFDM can efficiently utilize limited bandwidth channels allowing the use of advanced channel coding techniques. This combination facilitates a very robust communication over a power line channel.

Figure 6-2 shows the block diagram of an OFDM transmitter. The available bandwidth is divided into a number of sub-channels, which can be viewed as many independent PSK modulated subcarriers with different non-interfering (orthogonal) subcarrier 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 inverse fast Fourier transform (IFFT) on the complex-valued signal points produced by differentially encoded phase modulation that are allocated to individual subcarriers. An OFDM symbol is built by appending a cyclic prefix to the beginning of each block generated by IFFT. The length of a cyclic prefix is chosen so that the channel group delay does not cause excessive interference between successive OFDM symbols. Windowing reduces the out-of-band leakage of the transmit signals.

Channel estimation is used for link adaptation. Based on the quality of the received signal, the receiver (if requested by the transmitter) shall feed back the suggested modulation scheme to be used by the transmitting station in subsequent packets transmitted to the same receiver. Moreover, the system differentiates the subcarriers with insufficient SNR and does not transmit data on them.

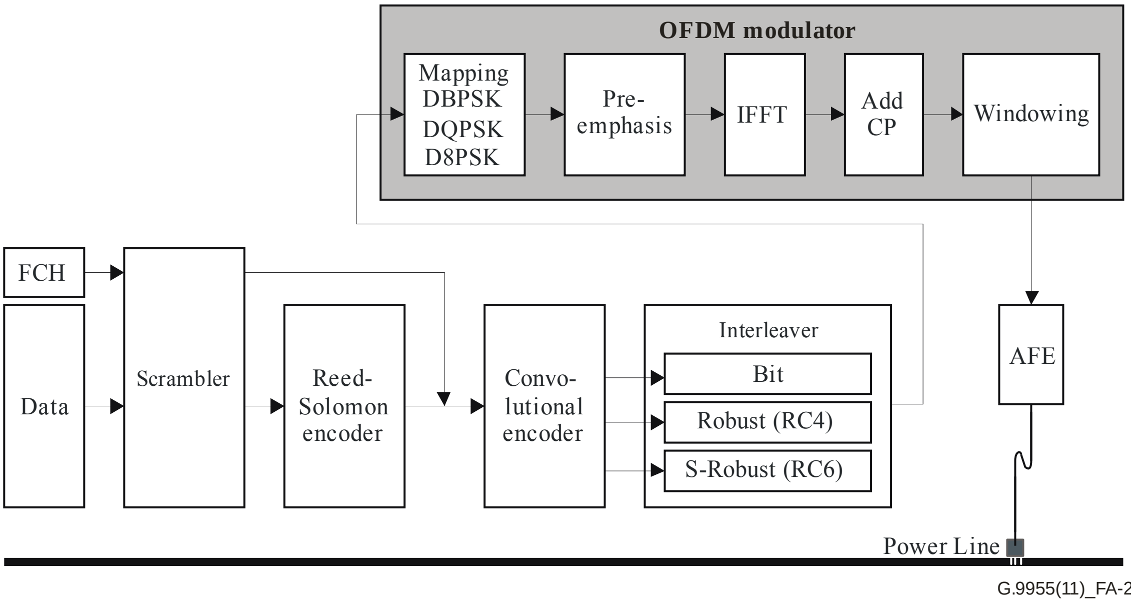


Figure 6-2 – Block diagram of an OFDM transceiver

7 Physical layer specification for the CENELEC-A bandplan

This clause specifies the physical layer block using the orthogonal frequency division multiplexing (OFDM) system in the CENELEC band.

7.1 Fundamental system parameters

ITU-T G.9903 devices support operation in the CENELEC-A band, as specified in Annex B of [ITU-T G.9901]. Mandatory values for the OFDM control parameters for the CENELEC-A bandplan are given in Table B.1 of [ITU-T G.9901]. The frequency band used for the CENELEC-A bandplan is defined in Table B.2 of [ITU-T G.9901].

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

As specified in Annex B of [ITU-T G.9901], the maximum number of subcarriers 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 subcarriers equal to 1.5625 kHz (Fs/N), where Fs is the sampling frequency and N is the IFFT size. Note that imperfection such as sampling clock frequency variation can 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 words, considering ±25 ppm sampling frequency in transmitter and receiver clocks, the drift of the subcarriers is approximately equal to 8 Hz that is approximately 0.5% of the selected frequency spacing. Considering these selections, the number of usable subcarriers is 36.

The system works in two different modes namely normal and robust modes. In normal mode, the FEC is composed of a Reed-Solomon encoder and a convolutional encoder. The system also supports Reed-Solomon code with a parity of 8 and 16 bytes.

In robust mode the FEC is composed of Reed-Solomon and convolutional encoders followed by a repetition code (RC). The RC code, repeats each bit four times making the system more robust to channel impairments. This of course will reduce the throughput by about a 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 acceptable robustness. The number of symbols, Reed-Solomon block sizes and data rate associated with 36 tones is tabulated in Tables 7-1 and 7-2.

Table 7-3 shows the rate including the data transmitted in the FCH. To calculate the data rate, it is assumed that the packets are continuously transmitted with no inter-frame time gap.

Table 7-1 – RS block size for various modulations

CENELEC-A Number of symbols	Reed-Solomon blocks (bytes) D8PSK (Out/In) (Note 1)	Reed-Solomon blocks (bytes) DQPSK (Out/In) (Note 1)	Reed-Solomon blocks (bytes) DBPSK (Out/In) (Note 1)	Reed-Solomon blocks (bytes) Robust (Out/In) (Note 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)
NOTE 1 – Reed-Solomon with 16 bytes parity. NOTE 2 – Reed-Solomon with 8 bytes parity.				

Table 7-2 – Data rate for various modulations (excluding FCH)

CENELEC-A	Data rate per modulation type, bit/s			
Number of symbols	D8PSK, P16[1]	DQPSK, P16[1]	DBPSK, P16[1]	Robust, P8[2]
12	21 829	12 619	3 410	N/A
20	32 534	20 127	7 720	N/A
32	42 619	27 198	11 778	N/A
40	N/A	30 385	13 608	2 423
52	N/A	33 869	15 608	3 121
56	N/A	34 792	16 137	3 257
112	N/A	N/A	20 224	4 647
252	N/A	N/A	N/A	5 592

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

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

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

Table 7-3 – Data rate for various modulations (including FCH)

CENELEC-A	Data rate per modulation type, bit/s			
Number of symbols	D8PSK, P16[1]	DQPSK, P16[1]	DBPSK, P16[1]	Robust, P8[2]
12	23 235	14 026	4 817	N/A
20	33 672	21 264	8 857	N/A
32	43 501	28 081	12 662	N/A
40	N/A	31 154	14 377	3 192
52	N/A	34 513	16 252	3 765
56	N/A	35 402	16 748	3 867
112	N/A	N/A	20 579	5 002
252	N/A	N/A	N/A	5 765

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

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

NOTE – N/A means not applicable and the reason for this is that the corresponding number of symbols specified results in an 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), number of subcarrier per symbol (N_{car}) and number of parity bits added by FEC blocks.

An example of how to calculate the data rate is given below using the following parameters:

• Number of FFT points	$N = 256$
• Number of subcarriers	$N_{car} = 36$
• Number of overlapped samples	$N_O = 8$
• Number of cyclic prefix samples	$N_{CP} = 30$
• Number of FCH symbols	$N_{FCH} = 13$
• Sampling frequency	$F_s =0.4$ MHz

- Number of symbols in preamble $N_{pre} = 9.5$

Consider the system in the CENELEC-A band working in the robust mode. The total number of bits carried by the whole PHY frame is equal to:

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

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

$$\text{No\_Bits\_Robust} = 1440 \times \text{Robust}_{Rate} = 1440 \times 1/4 = 360 \text{ bits}$$

Considering the fact that the convolutional encoder has a rate equal to 1/2 ($CC_{Rate} = 1/2$) 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 ($MAXRS_{bytes}$) shall be equal to:

$$MAXRS_{bytes} = \text{floor}((\text{No\_Bits\_Robust} \times CC_{Rate} - \text{CCZeroTail})/8) = \text{floor}((360 \times 1/2 - 6)/8) = 21$$

Removing 8 bytes associated with the parity bits (in robust mode) we obtain:

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

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

$$T_{Frame} = (((N_S + N_{FCH}) \times (N_{CP} + N - N_O) + (Npre \times N)))/Fs$$

Where N_{pre}, N, N_O and N_{CP} are the number of symbols in the preamble, FFT length, the number of samples overlapped at each side of one symbol and the number of samples in the cyclic prefix, respectively. N_{FCH} is the number of symbols in the FCH. The F_s is the sampling frequency.

Substituting the above numbers in the equation, T_{Frame} (PHY frame duration) for a 40-symbol frame is obtained as follows:

$$T_{Frame} = ((40 + 13) \times (256 + 22) + (9.5 \times 256)) / 400000 = 0.043 \text{ s.}$$

Therefore the data rate is calculated by:

$$\text{Data rate} = 104/0.042 \sim 2.4 \text{ kbit/s}$$

7.2 Frame structure

The PHY supports two types of frames. A typical data frame for the OFDM PHY is shown in Figure 7-1. Each frame starts with a preamble which is used for synchronization and detection in addition to AGC adaptation. SYNCP simply refers to symbols that are multiplied by +1 in the sign function above, 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. The preamble is followed by 13 data symbols allocated to the frame control header (FCH). The FCH has the important control information required to demodulate the data frame. Data symbols are transmitted next. The first FCH symbol uses the phase from the last preamble P symbol and the first data symbol uses the phase from the last FCH symbol. In the figures, "GI" stands for guard interval, which is the interval containing the cyclic prefix.

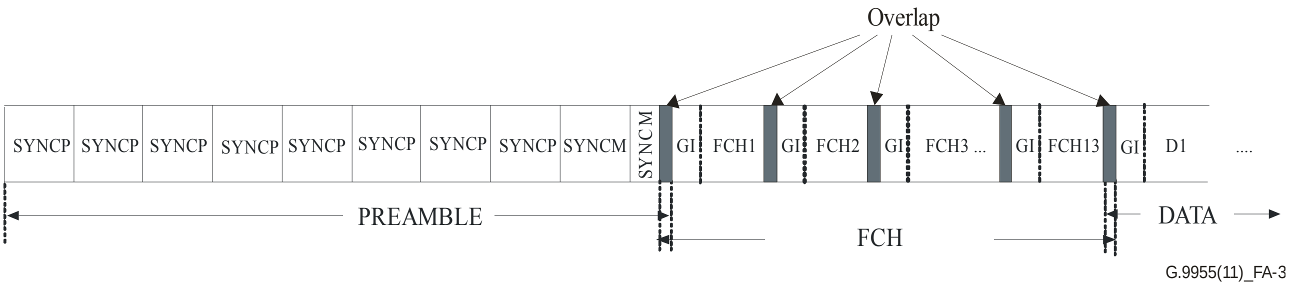


Figure 7-1 – Typical data frame structure

The PHY also supports an ACK/NACK frame which only consists of the preamble and the FCH. The frame structure of the ACK frame is shown in Figure 7-2. The bit fields in the FCH, explained in clause 7.4, will perform the ACK/NACK signalling.

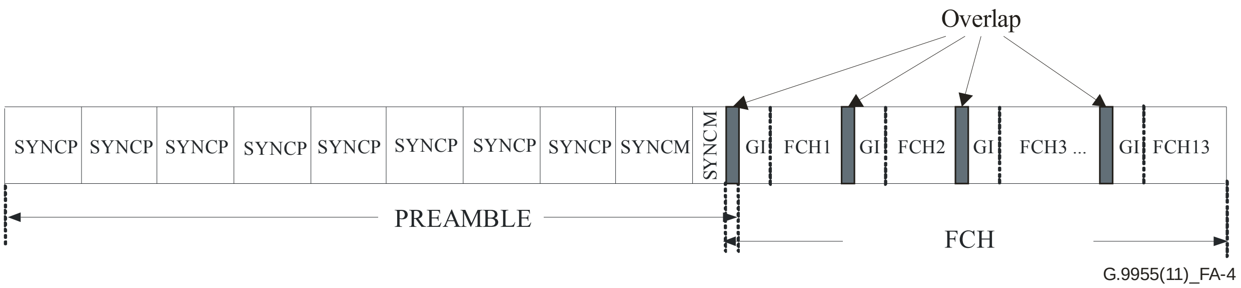


Figure 7-2 – ACK/NACK frame structure

7.3 Preamble

The preamble is composed of 8 identical SYNCP symbols and 1½ identical SYNCM symbols. Each of the SYNCP and SYNCM symbols is 256 samples and is pre-stored in the transmitter and transmitted right before the data symbols. The SYNCP symbols are used for AGC adaptation, symbol synchronization, channel estimation and initial phase reference estimation. The SYNCM symbols are identical to the SYNCP symbols except that all the subcarriers are π phase shifted. At the receiver, the phase distance between symbol SYNCP and symbol SYNCM waveforms is used for frame synchronization. A SYNCP symbol is generated by creating 36 equally spaced subcarriers with the phase of each subcarrier given by ϕ_c as shown in Table 7-4. One way to generate this signal is to start in the frequency domain and create 36 complex subcarriers with the initial phases ϕ_c, as shown in Table 7-4. Figure 7-12 shows how the 36 subcarriers are mapped to the IFFT input where the first modulated subcarrier is subcarrier 23 and the last modulated subcarrier is subcarrier 58.

Table 7-4 – Phase vector definition

c	ϕ_c	c	ϕ_c	c	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(π/8)	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)

Table 7-4 – Phase vector definition

c	ϕ_c	c	ϕ_c	c	ϕ_c
7	7(π/8)	19	15(π/8)	31	3(π/8)
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)

7.4 Frame control header

The thirteen data symbols immediately after the preamble are reserved for the frame control header (FCH). 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 the frame, tone map index of the frame, length of the frame, etc. The FCH data is protected with CRC5. Table 7-5 defines the structure of the FCH. The FCH shall use the default tone map (all allowed subcarriers).

The tone map (see clause 11.3.3.2.2) field of the FCH is made of 9 bits, numbered from TM[0] to TM[8], where TM[7] is the most significant bit (MSB) of one byte while TM[0] is the least significant bit of that byte; TM[8] is the MSB of the second byte. These nine bits are mapped to frequency bands as in the following:

– TM[8]: Unused in CENELEC-A band

– TM[7]: Unused in CENELEC-A band

– TM[6]: Unused CENELEC-A band

– TM[5] is 82.8125 to 90.625 kHz

– TM[4] is 73.4375 to 81.25 kHz

– TM[3] is 64.0625 to 71.875 kHz

– TM[2] is 54.6875 to 62.5 kHz

– TM[1] is 45.3125 to 53,125 kHz

– TM[0] is 35.9375 to 43.75 kHz.

Table 7-5 – FCH bit fields

Field	Byte	Bit number	Bits	Definition
PDC	0	7-0	8	Phase detection counter
MOD	1	7-6	2	Modulation type: 00: Robust mode (clause 7.6.3) 01: DBPSK 10: DQPSK 11: D8PSK
FL	1	5-0	6	PHY frame length in PHY symbols
TM[7:0]	2	7-0	8	TM[7:0] – Tone map
TM[8]	3	7	1	TM[8] – Tone map

Table 7-5 – FCH bit fields

Field	Byte	Bit number	Bits	Definition
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 by ITU-T
FCCS	3	3-0	4	Frame control check sequence (CRC5)
	4	7	1	
ConvZeros	4	6-1	6	6 zeros for convolutional encoder
NOTE – The robust mode uses DBPSK with 4 repetitions.				

The frame length bit field gives the number of symbols in the frame based on the formula:

Number of symbols = FL × 4

A 5-bit cyclic redundancy check (CRC) is used for error detection in the FCH. The CRC5 is calculated using the following standard generator polynomial of degree 5:

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

7.4.1 Data

The data to transport in a physical frame (PSDU) is provided by the upper layer as a byte stream and is read most significant bit first into the scrambler. The upper layer shall be responsible for padding the data to accommodate the requirement of the PHY layer (see clause I.1).

7.5 Scrambler

The data scrambler block helps give the data a random distribution. The data 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 7-3. The bits in the scrambler are initialized to all-ones at the start of processing each PHY frame.

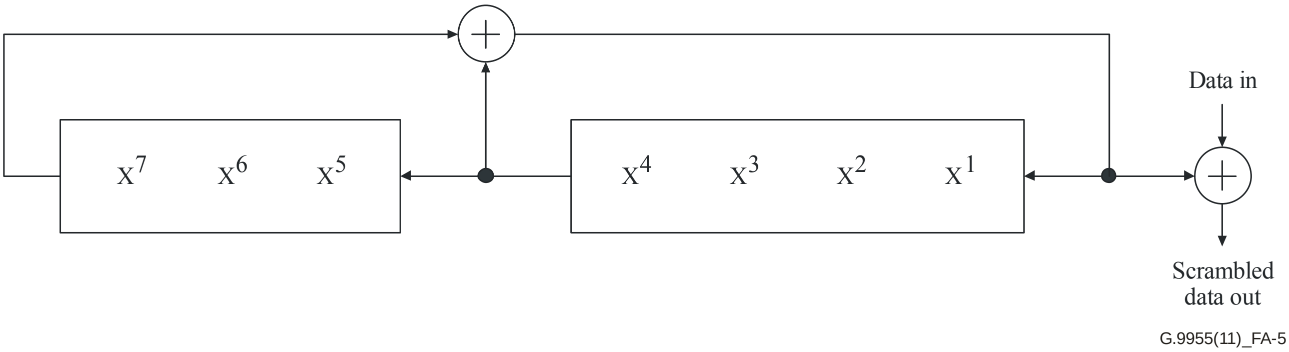


Figure 7-3 – Data scrambler

7.6 FEC coding

The FEC encoder is composed of a Reed-Solomon encoder followed by a convolutional encoder. In robust mode, an encoder, namely, repetition code (RC4), is used after the convolutional encoder in order to repeat the bits at the output of convolutional encoder four times. In super robust mode, an encoder, namely, repetition code (RC6), is used after the convolutional encoder in order to repeat the bits at the output of the convolutional encoder six times.

7.6.1 Reed-Solomon encoder

For the data portion of a frame, data from the scrambler is encoded by shortened systematic codes using Galois field GF(2^8). Only one RS block is used by a frame. Depending on the mode used the following parameters are applied:

- Normal mode: RS(N = 255, K = 239, T = 8)
- Robust mode: RS(N = 255, K = 247, T = 4)

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 value of T (number of correctable symbol errors) can be either 4 or 8 for different configurations. For the robust mode, the code with T=4 is used. The number of parity words in an RS-block is 2T bytes.

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

Field generator polynomial: $p(x) = x^8 + x^4 + x^3 + x^2 + 1$ (435 octal)

The representation of α0 is "00000001", where the left most bit of this RS symbol is the MSB and is the 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 α^1 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 $d^7\alpha^7 + d^6\alpha^6 ... +d^1\alpha + d^0$.

The first bit in time from the data scrambler becomes the most significant bit of the symbol at the input of the RS encoder. Each RS encoder input block is formed by one or more fill symbols ("00000000") followed by the message symbols. Output of the RS encoder (with fill symbols discarded) proceeds in time from the first message symbol to the last message symbol followed by parity symbols, with each symbol shifted out most significant bit first.

7.6.2 Convolutional encoder

The bit stream at the output of the Reed-Solomon block is encoded with a standard rate =1/2, K=7 convolutional encoder. The tap connections are defined as x = 0b1111001 and y = 0b1011011, as shown in Figure 7-4.

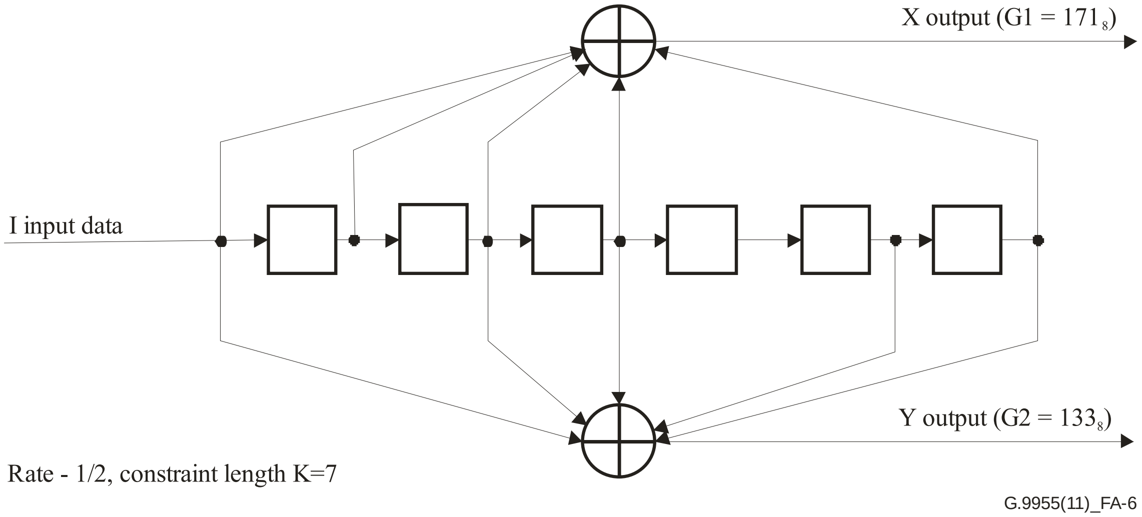


Figure 7-4 – Convolutional encoder

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 improves the error probability of the convolutional decoder, which relies on future bits when decoding. The tail bits are defined as six zeros.

Zero bit padding is used to fit the encoded bits into a number of OFDM symbols that is a multiple of 4. The location of the bit padding shall be at the end of the convolutional encoder output and, in case of the robust mode, the bit padding is done before the repetition block.

7.6.3 Robust and super robust modes

When robust or super robust modes are used, the underlying modulation is always DBPSK. Robust and super robust modulation, provide extensive time and frequency diversity to improve the ability of the system to operate under adverse conditions.

7.6.3.1 Repetition coding by 4 (RC4)

In robust mode, every bit at the output of the convolutional encoder is repeated four times and then passed as input to the interleaver as described in clause 7.7. This encoder (RC4) is only activated in robust mode.

7.6.3.2 Repetition coding by 6 (RC6)

In the super robust mode, every bit at the output of the convolutional encoder is repeated six times and then passed as input to the interleaver as described in clause 7.7. Only the FCH uses the super robust mode but without Reed-Solomon encoding.

7.7 Interleaver

The interleaver is designed as such so that it can provide protection against two different sources of error:

- 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.

We define m as the number of used data carriers in each OFDM symbol, n as the number of OFDM symbols used by the frame and total_number_of_bits as the total number of coded bits including the padding bits.

$$n = ceil\left(\frac{Total\_number\_of\_bits}{4 \times m \times mod_{size}}\right) \times 4$$

with mod_size=1, 2, 3, 4 is the modulation size, i.e., the number of bits per constellation symbol.

From m and n the circular shift parameters m_i, m_j, n_i and n_j are derived.

To get a proper parameter set, m_i, m_j, n_i and n_j should be the smallest figures to comply with these conditions:

- GCD(m_i, m) = GCD(m_j, m) = 1
- m_i < m_j
- GCD(n_i, n) = GCD(n_j, n) = 1
- n_j < n_i

These parameters form an elementary permutation matrix (dimensions are m columns and n rows) taking input bits from their original position to the interleaved position following the formula below:

$$J = (j \times n\_j + i \times n\_i) \% n$$

$$I = (i \times m\_i + J \times m\_j) \% m$$

where

(i,j) are the original bit position (i = 0, 1,..., m-1 and j = 0, 1,..., n-1)

and

(I,J) are their corresponding interleaved position.

The DBPSK modulation permutation matrix corresponds to the elementary permutation matrix while DQPSK and D8PSK modulations use respectively two and three times the elementary permutation matrix. Thus, the dimension of the permutation matrix for DQPSK and D8PSK modulations are m columns and n×mod_size.

The data to be interleaved are stored in the input buffer and shows which dimensions are m columns and n×mod_size rows.

The data bits are put in the input buffer row by row as shown in Figure 7-5. Zero padding will be used to match permutation matrix dimensions.

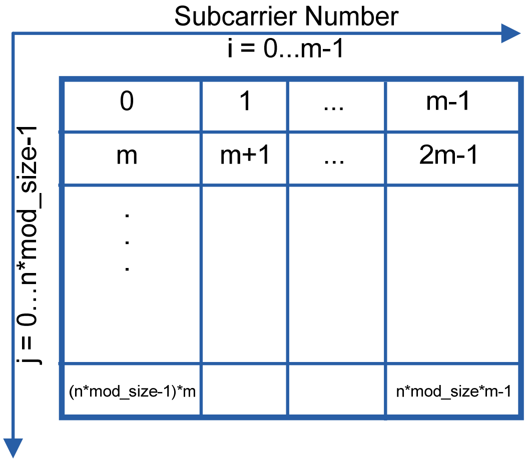


Figure 7-5 – Bit Order input into the input buffer

Once interleaved each bit is stored in an output buffer as shown in Figure 7-6.

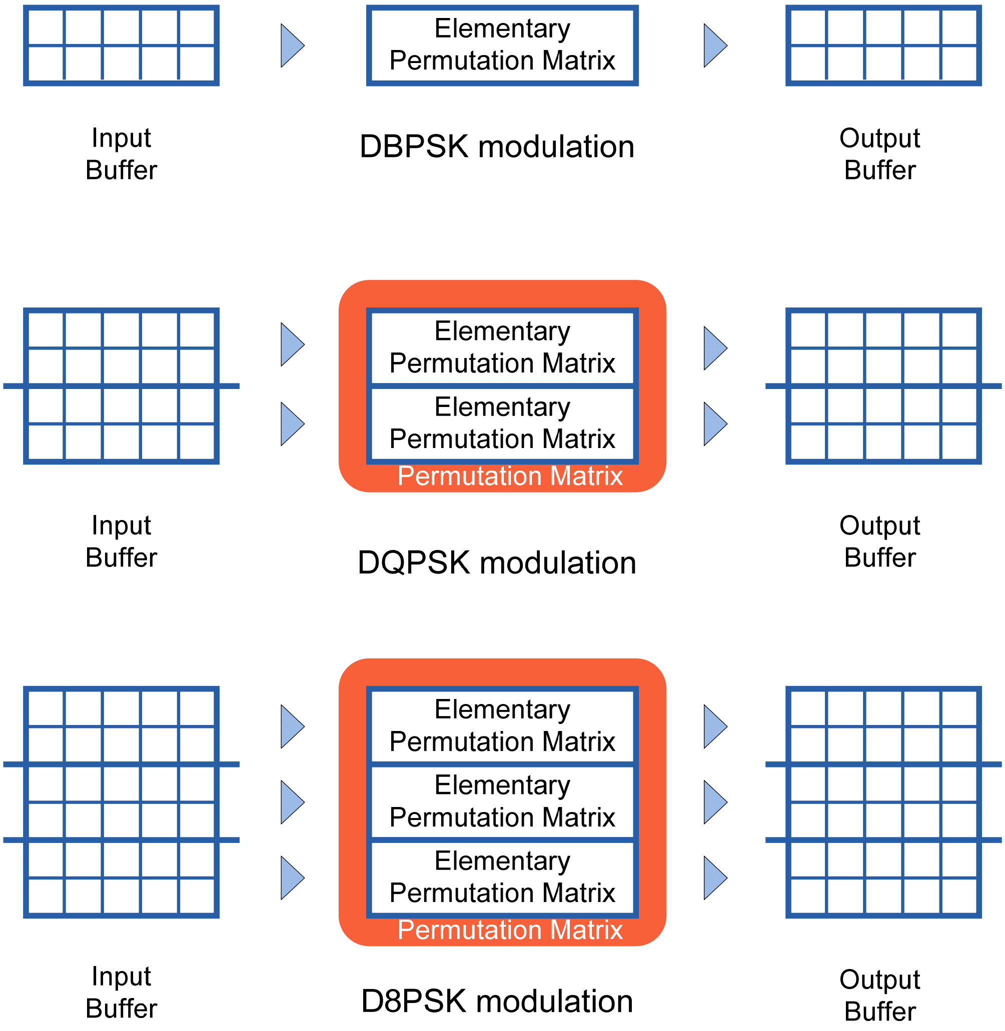


Figure 7-6 – Permutation matrix used with different modulations

After interleaving, the mapping functions used for modulation read the output buffer row by row. Each sequence of mod_size bit(s) is (are) computed to form a symbol.

An example is given here for information purposes.

A simple search is done to find a good set of parameters based on m and n.

For a given value of n, n_j shall be the first co-prime larger than 2 and n_i shall be the second co-prime larger than 2. Similarly, for a given value of m, m_i shall be the first co-prime larger than 2 and m_j shall be the second co-prime larger than 2. Figure 7-7 displays the spreading behaviour of the interleaver for n = 8, m = 10, n_i = 5, n_j = 3, m_i = 3 and m_j = 7.

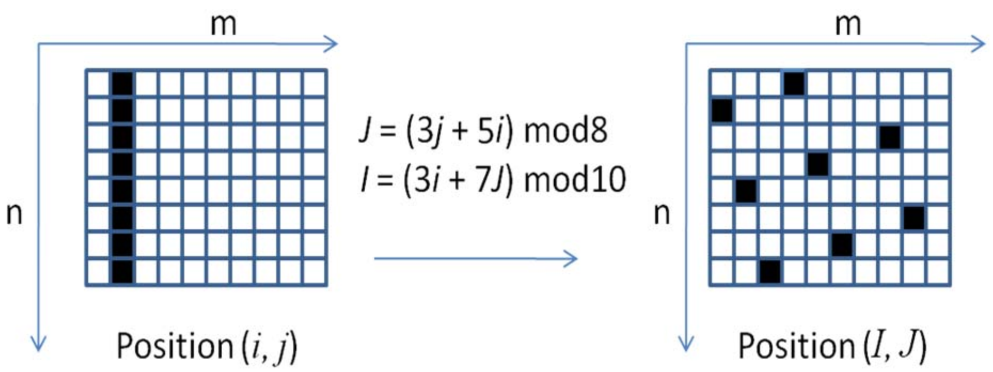


Figure 7-7 – Example of spreading behaviour

The calculation of n_i, n_j, m_i and m_j are explained as below:

- n = 8 (co-prime numbers for 8 except 1 and 2 are: 3, 5, 7). The first number is 3, so n_j = 3; and the next co-prime with 8 is 5, so n_i= 5; that is the first co-prime number other than 1 and 2 of n shall be n_j, and the second co-prime of n other than 1 and 2 shall be n_i;
- m = 10 (co-prime numbers for 10 except 1 and 2 are: 3, 7, 9). The first number we meet in the set is 3, so m_i=3; and the next is 7, so m_j=7; that is the first co-prime of m other than 1 and 2 shall be m_i, and the next co-prime shall be m_j.

Here, we use DBPSK and DQPSK as examples. Suppose we have 3 active tones (m=3) and 2 symbols (n=2).

With DBPSK modulation:

If the input bit stream is "123456", the input bit stream will be loaded into the matrix as Figure 7-8(a). The vertical dimension of the matrix is n×mod_size (i.e., 2×1=2). After that, interleaving is done with interleaving block size n×m (i.e., 2×3). After all the bits have been processed, the bits 1'2'3'…6' are mapped to the modulator as shown in Figure 7-8(c).

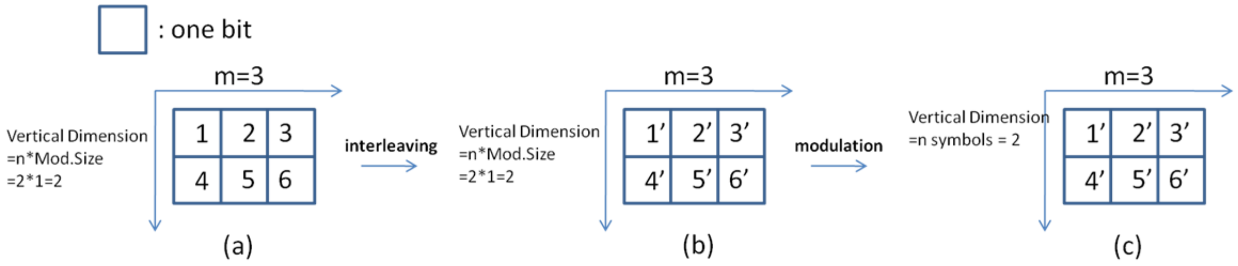


Figure 7-8 – Example of interleaving with DBPSK

With DQPSK modulation:

If the input bit stream is "1 2 3 4 5 6 … 12", the input bit stream will be loaded into the matrix as Figure 7-9(a). The vertical dimension of the matrix is n×mod_size (i.e., 2×2=4). After that, interleaving is done with interleaving block size n×m (i.e., 2×3). After all the bits have been processed, the bits 1' 2' 3'…11' 12' are mapped to the modulator as shown in Figure 7-9(c).

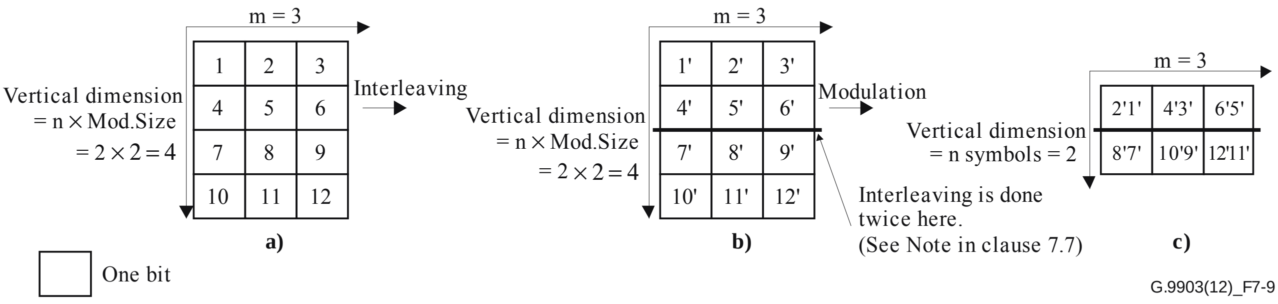


Figure 7-9 – Example of interleaving with DQPSK

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

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

where the interleaving table ILV_TBL and the interleaving size ILV_SIZE are defined as follow:

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;
    }
}
```

NOTE – For the above DBPSK example, ILV_SIZE = m × n = 3 × 2 = 6, size = 3 × 2 = 6, so the loop runs once. For the above DQPSK example, ILV_SIZE = m × n = 3 × 2 = 6, size = 3 × 4 = 12, so the loop runs twice.

7.8 DBPSK/DQPSK/D8PSK mapping

Each subcarrier is modulated with differential binary or differential quadrature phase shift keying (DBPSK or DQPSK or D8PSK) or robust modulation. Forward error correction (FEC) is applied to both the frame control information (super robust encoding) and the data (concatenated Reed-Solomon and convolutional encoding) in the communication packet.

The mapping block is also responsible for assuring that the transmitted signal conforms to the given tone map and tone mask. The tone map and mask are concepts of 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 subcarriers that are to be used for a particular communication between two modems. For example, subcarriers that suffer deep fades can be avoided and no information is transmitted on those subcarriers.

7.8.1 Mapping for DBPSK, DQPSK, D8PSK modulations

Data bits are mapped for differential modulation (DBPSK, DQPSK, D8PSK). Instead of using the phase reference vector ϕ, each phase vector uses the same subcarrier, previous symbol, as its phase reference. The first FCH symbol uses the phase from the last preamble P symbol and the first data symbol uses the phase from the last FCH symbol. The data encoding for DBPSK DQPSK and DQPSK is defined in Tables 7-6, 7-7 and 7-8, where Ψk is the phase of the k-th subcarrier from the previous symbol. In DBPSK a phase shift of 0 degrees represents a binary "0" and a phase shift of 180 degrees represent a binary "1". 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. 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, 011, 010, 110, 111, 101 and 100 respectively.

Table 7-6 – DBPSK encoding table of k-th subcarrier

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

Table 7-7 – DQPSK encoding table of k-th subcarrier

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

Table 7-8 – D8PSK encoding table of k-th subcarrier

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

Alternatively, the phase differences used to compute "output phases" in Tables 7-6 and 7-7 can be represented in a constellation diagram (with reference phase assumed equal to 0 degrees), as shown in Figure 7-10.

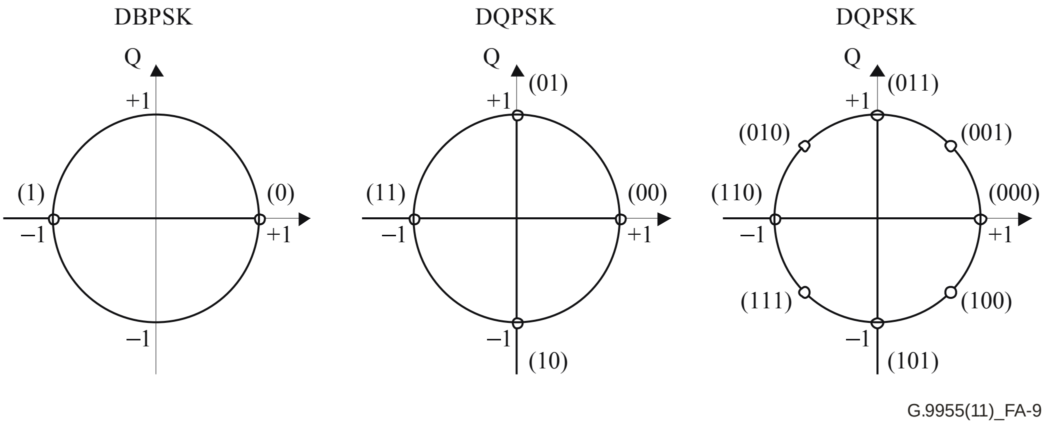


Figure 7-10 – Constellation encoding

7.9 Frequency domain pre-emphasis

The purpose of this block is to provide frequency shaping to the transmit signal in order to compensate for attenuation introduced to the signal as it goes 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 clause 8.6. 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 as described in clause 7.12. 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 7-11 below shows a block diagram of the pre-emphasis filter. The output of the filter shall be the input to the IFFT.

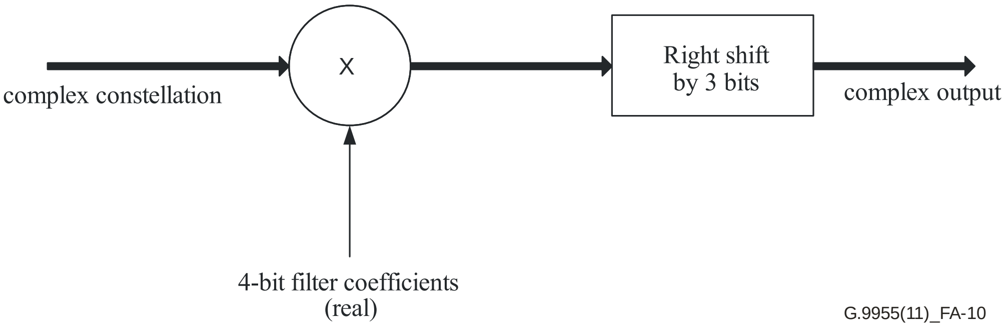


Figure 7-11 – Block diagram of the pre-emphasis filter

7.10 OFDM generation (IFFT and CP addition)

The OFDM signal can be 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. In other words, we take the last 30 samples at the output of the IFFT and place them in front of symbol. The useful output is the real part of the IFFT coefficients. The input/output configuration is as depicted in Figure 7-12.

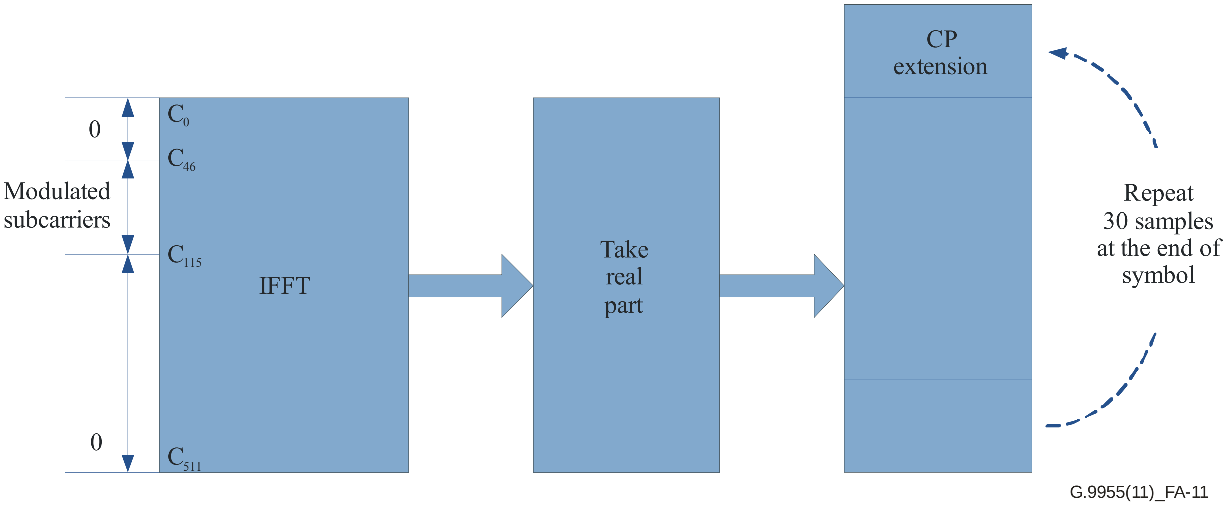


Figure 7-12 – IFFT input/output and CP addition

7.11 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 7-13.

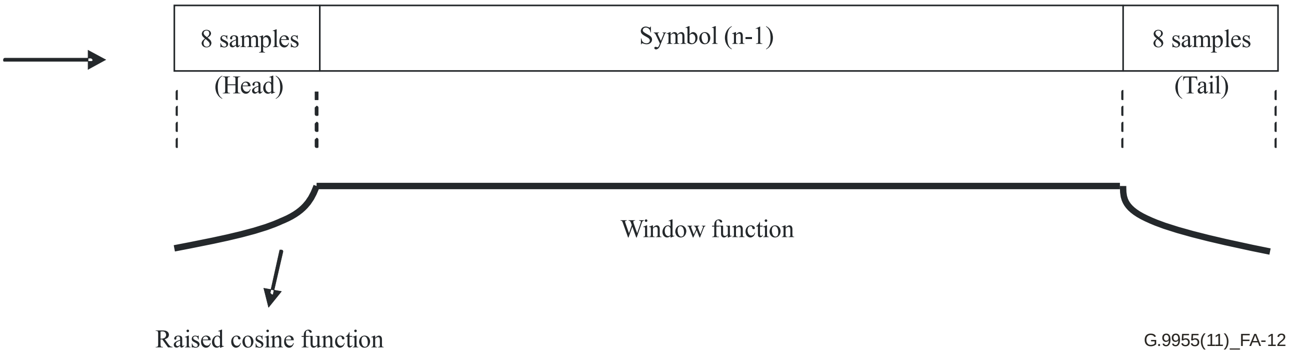


Figure 7-13 – Raised cosine windowing

The windowing function at each 8-sample boundary is a raised cosine function and its values are given in Table 7-9. 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 the symbol are overlapped with the tail and head samples of adjacent symbols as shown in Figure 7-14.

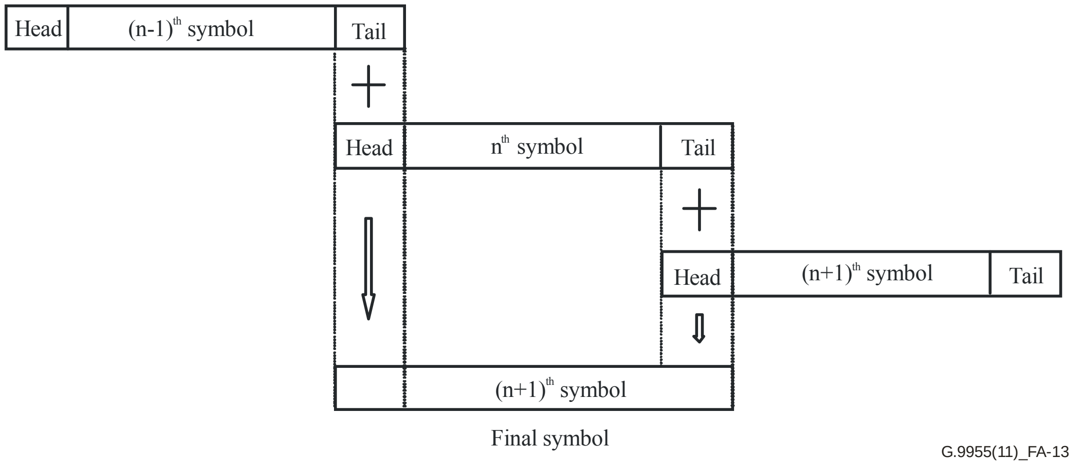


Figure 7-14 – Overlap/add

Figure 7-14 – In other words, in order to construct the n-th symbol, first its 8 head samples are overlapped with the 8 tail samples of the (n–1)-th symbol and its 8 tail samples are overlapped with the 8 head samples of the (n+1)-th symbol. Finally, the corresponding overlapped parts are added together. Note that the head of the first symbol is overlapped with the tail of the preamble. And the tail of the last symbol is sent out with no overlapping applied.

Table 7-9 – The raised cosine samples

	Head samples	**Tail samples**
1	0	0.9619
2	0.0381	0.8536
3	0.1464	0.6913
4	0.3087	0.5000
5	0.5000	0.3087
6	0.6913	0.1464
7	0.8536	0.0381
8	0.9619	0

7.12 Adaptive tone mapping and transmit power control

ITU-T G.9903 devices shall estimate the SNR of the received signal subcarriers and adaptively select the usable tones and optimum modulation and code rate (including DBPSK, DQPSK, D8PSK) to ensure reliable communication over the power line channel. It shall also specify which power level the remote transmitter shall use and which gain values it should apply for various sections of the spectrum. The per-tone quality measurement enables the system to adaptively avoid transmitting data on subcarriers 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 group of subcarriers will be used for data transmission and which ones will be used to send dummy data that the receiver shall ignore. However, at least one group of subcarriers (as indicated by the TM field of the FCH – see clause 7.4) shall carry data.

The goal of the adaptive tone mapping is to allow the ITU-T G.9903 receiver to achieve the greatest possible throughput given the channel conditions existing between them. 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 source station may request a destination station to estimate a channel condition by setting the TMR bit of the FCH as described in clause 7.4.

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

The parameters of the tone map response message are shown in Table 11-9.

7.12.1 PN modulating unused subcarriers

For the data part of the frame, the mapping function for DBPSK, DQPSK, D8PSK and "Robust" shall obey the tone map; thus subcarriers 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 subcarrier is encountered on which no information is to be transmitted, the mapping function substitutes a binary value from a pseudo noise (PN) sequence. The one bit output of the PN sequence should be duplicated for all modulated bits (1 for BPSK, 1x2 for QPSK, 1x3 for 8PSK, etc.).

The PN sequence shall be generated using the same generator polynomial introduced in clause 7.5. 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 35th value corresponds to carrier number 0 of the second OFDM symbol.

7.13 Crossing MV/LV transformer

ITU-T G.9903 devices operate over both low-voltage and medium-voltage power lines. When operating over a medium-voltage power line an ITU-T G.9903 device can communicate with ITU-T G.9903 devices operating over low-voltage power lines. This means that the receiver on the LV side can detect the transmitted signal after it has been severely attenuated as a result of going through an MV/LV transformer. As the signal goes through the transformer, it is expected to experience overall severe attenuation in its power level as well as frequency-dependent attenuation. Both the transmitter and receiver have mechanisms to compensate for this attenuation. The transmitter can adjust its overall signal level as well as shape its power spectrum, while the receiver has an automatic gain control in order to achieve enough gain to compensate for the overall attenuation.

An ITU-T G.9903 node, in addition to being able to operate in normal mode, can operate as a repeater. When configured in "repeater" mode, the ITU-T G.9903 node can decode received frames and then retransmit them at a higher signal level in order to partially compensate for the attenuation introduced by the transformer. The repeater, when needed, can be placed on the LV side of the MV/LV transformer.

7.14 MV coupler (informative)

ITU-T G.9903 devices interface with the MV power line through a PLC coupling device, which is basically a high-pass filter whose purpose is to permit the PLC signal to pass, but reject the power system frequency and protect the communications equipment from the power system voltage and transient voltages caused by switching operations.

The basic circuit diagram is shown in the figure below. A complete coupling comprises a line trap to prevent the PLC signal from being short-circuited by the substation, and a coupling filter formed by the coupling capacitor and the coupling device.

For ITU-T G.9903 devices, resolving impedance mismatching is very important in the sense of transferring maximum power to the signal input terminal of the MV power distribution lines. It is recommended that any transformer being used should be verified by measuring transmission and reflection characteristics through a vector network analyser.

The proposed coupling interface, shown in Figure 7-15, should interface between the PLC device and the MV medium (with 24 kV and impedance of 75 Ω to 175 Ω).

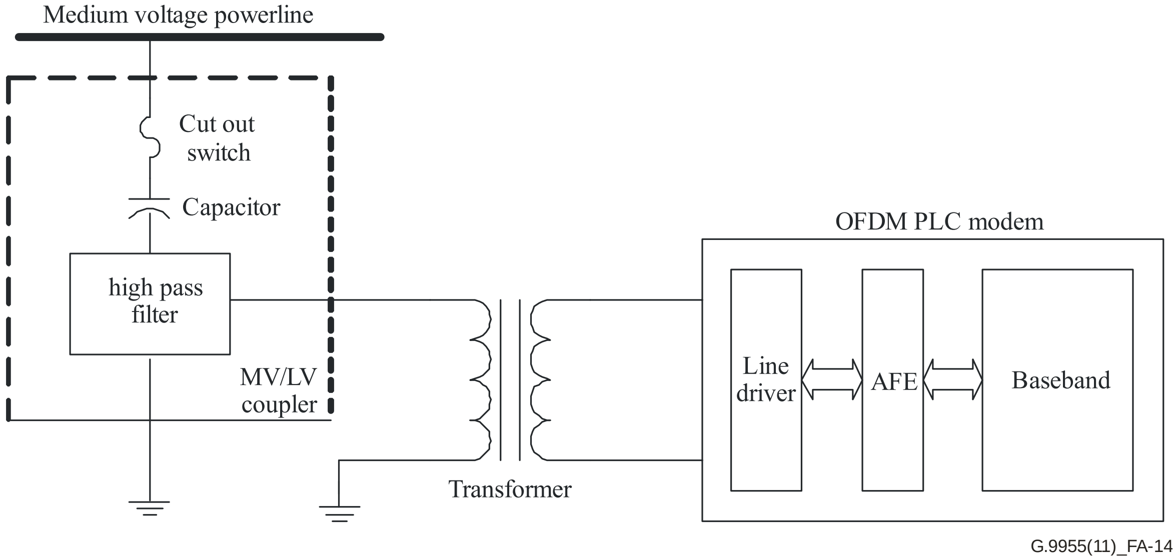


Figure 7-15 – Proposed coupling circuit

7.14.1 Coupler technical characteristics (informative)

Table 7-10 – Coupler technical characteristics

Parameter	Measurement conditions	Value
	Medium-voltage circuit parameters	
Primary test voltage U_N	Voltage between the device input and grounding output	$24/\sqrt{3}$ kV_{rms}
Test short-term alternating voltage U_{TH}	Voltage between the device input and grounding output during one minute	50 kV_{rms}
Maximum short-term working voltage U_{MAX}	Medium voltage during nine hours	26 kV_{rms} 9 hours
Test lightning impulse voltage U_L	Impulse with duration of 1,2/50 us between the device input and the grounding output	125 kV
Partial discharge level		≤ 20 pC

Table 7-10 – Coupler technical characteristics

Parameter	Measurement conditions	Value
Ambient temperature during operation		–40°C - +65°C
Coupling capacitor capacity Cc	–40°C < Ta < +70°C	1.5 nF - 13 nF
Fuse operate time max	at I ≥ 30 A at I ≥ 45 A	t ≤ 100 ms t ≤ 10 ms
	Low-voltage circuit parameters	
Nominal line side impedance R_{LINE}		75 Ω ≤ R ≤ 170 Ω
Nominal equipment side impedance R_{LOAD}		75 Ω
Maximum operating attenuation in receive and transmit direction at R_{LOAD} = 75 Ω, R_{LINE} = 170 Ω	35 kHz ≤ f ≤ 170 kHz	3 dB

7.15 AC phase detection

It is necessary to know which phase each meter is placed on in an AMM application. This information is mainly useful at the system level in order to check for unexpected losses on the distribution line and shall be stored in the MIB.

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 transition through zero volts of a 50 Hz sinusoidal on a power line and shall be used to synchronize a Tx-meter and an Rx-meter. The Tx-meter generates a time stamp based on internal counter at the instant a packet shall be transmitted. The receiver provides its own time stamp and delay between the Tx-meter and the Rx-meter provides the phase difference. The procedure to achieve the phase difference between the transmitter and receiver is as follows:

1) All devices including the meter and data concentrator shall have an internal timer, which are synchronized with the zero-crossing detector.

2) All devices shall have a zero-crossing detector which delivers an output pulse so that the pulse width is 5% of the total period. The characteristic of the zero-crossing detector is shown in Figure 7-16.

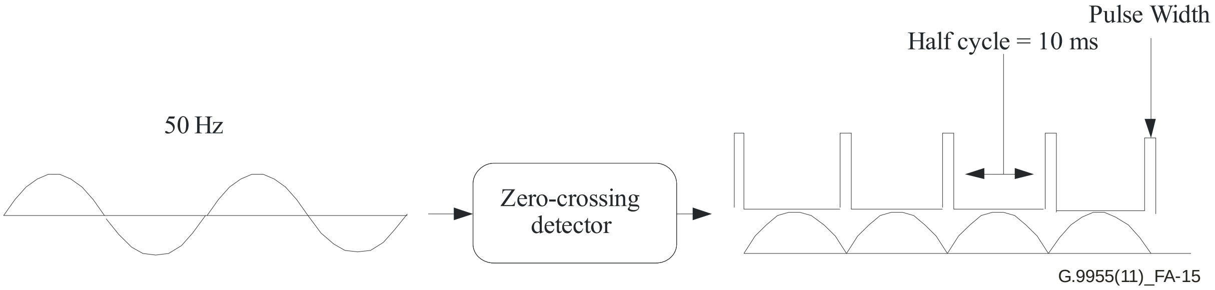


Figure 7-16 – Zero-crossing detector

3) An eight bits counter provides a time stamp placed on the FCH frame upon transmission of the payload. This counter counts from zero to 255 in one period of the mains and is reinitialized each time a zero-crossing event is detected.

4) Upon detection of an FCH frame, the receiver shall compute the delay, which is the difference between a transmit counter and a received counter. The phase differential shall be computed as shown below.

$$\text{Phase differential} = (\text{Rx\_counter} - \text{Tx\_Counter})/3$$

Electromagnetic propagation time and additional delay for packet processing and detection shall be considered a measuring delay. An electromagnetic propagation delay is 5.775 us/km, which can be neglected; however, a processing delay shall be factored into the equation above as follows:

$$\text{New\_Phase differential} = (\text{Rx\_counter} - \text{detection\_delay}) - (\text{Tx\_Counter} - \text{transmission\_delay})/3$$

8 Transmitter electrical specifications

8.1 Output level measurement

See the main body of [ITU-T G.9901].

8.2 Transmit spectrum mask

See clause B.2 of [ITU-T G.9901].

8.3 Spurious transmission

See clause B.2.1 of [ITU-T G.9901].

8.4 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.

8.5 Transmit constellation accuracy

8.5.1 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.

8.5.2 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 at 400 K samples per second or more, 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:

1) Pass a sequence of 37 bytes all-ones, representing a 12-symbol QPSK frame, through an ideal floating-point transmitter and save the complex input to the IFFT block for each of the 12 data symbols as $A_{i,c}e^{j\Phi_{i,c}}$, where $A_{i,c}e^{j\Phi_{i,c}}$ is the reference constellation point corresponding to the -th OFDM symbol carried over the *c*-th subcarrier. Index '*i*' shall have values between 0 and 11 while index '*c*' shall be between 0 and 35. The ideal transmitter should include all the transmitter blocks specified in this Recommendation, including scrambler, forward error correction, interleaver and mapper.

2) Next, use the transmitter under test to generate the same frame using the bits specified in step 1.

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

4) Save all time sample of the 12 OFDM symbols of the frame.

5) Offline, apply a floating point FFT on each OFDM symbol and store the complex values as $B_{i,c}e^{j\Theta_{i,c}}$ where ' ' is the OFDM symbol number and '*c*' is the carrier number corresponding to that symbol. $B_{i,c}e^{j\Theta_{i,c}}$ represents the actually transmitted constellation point and, ideally, $A_{i,c}e^{j\Phi_{i,c}} = B_{i,c}e^{j\Theta_{i,c}}$.

6) Compute the mean square error (MSE) between the ideal constellation points and the actually transmitted ones obtained at the end of step 5 for each symbol as the sum of the squared Euclidean distance between the two points over all the subcarriers in the symbol. The MSE of the *i*-th symbol is defined as:

$$MSE_i = \frac{1}{36}\sum_{c=0}^{35}\left|A_{i,c}e^{j\Phi_{i,c}} - B_{i,c}e^{j\Theta_{i,c}}\right|^2$$

Next, compute the total MSE as the sum of the MSEs of the each OFDM symbols:

$$Total\_MSE = \sum_{i=0}^{11} MSE_i$$

7) Compute the average energy of the reference constellation points carried by the *i*-th OFDM symbol:

$$\mathrm{Avg\_En}_i^{(ref)} = \frac{1}{36}\sum_{c=0}^{35}\left|A_{i,c}\right|^2$$

and the total average energy for all transmitted OFDM symbols as:

$$\mathrm{Tot\_En}^{(ref)} = \sum_{i=0}^{11}\mathrm{Avg\_En}_i^{(ref)}$$

The normalized total MSE in dB should satisfy the following equation:

$$10\log_{10}\left(\frac{Total\_MSE}{Tot\_En^{(ref)}}\right) < -15\,\mathrm{dB}$$

8.6 Transmitter spectral flatness

See clause B.2.2 of [ITU-T G.9901].

9 PHY primitives

9.1 Data primitive

The receipt of the PD-DATA.request primitive by the PHY entity will cause the transmission of the supplied PSDU to be attempted. The PHY will first construct a PHY protocol data unit (PPDU) containing the supplied PSDU, and then transmit the PPDU. If the PD-DATA.request primitive is received by the PHY while the receiver is not enabled, or the transmitter is busy transmitting, the PHY shall first construct a PPDU containing the supplied PSDU, and then transmit the PPDU. When the PHY entity has completed the transmission successfully, it shall issue the PD-DATA.confirm primitive with a status of SUCCESS. If a PD-DATA.request primitive is received while the receiver is enabled (TXOFF_RXON state), the PHY entity shall discard the PSDU and issue the PD-DATA.confirm primitive with a BUSY_RX status. If a PD-DATA.request primitive is received while the transmitter is already busy transmitting (BUSY_TX state), the PHY entity shall discard the PSDU and issue the PD-DATA.confirm primitive with a BUSY_TX status.

If the processing or transmission of PHY is not possible due to invalid parameters or for any other reason, the PHY entity shall discard the PSDU and issue the PD-DATA.confirm primitive with a FAILED status.

The receipt of the PD-ACK.request primitive by the PHY entity will cause the transmission of the ACK/NACK frame to be attempted. The PHY will first construct an ACK/NACK frame and then transmit it. When the PHY entity has completed the transmission successfully, it shall issue the PD-ACK.confirm primitive with a SUCCESS status. If a PD-ACK.request primitive is received while the receiver is enabled (TXOFF_RXON state), the PHY entity shall discard the constructed ACK/NACK frame and issue the PD-ACK.confirm primitive with a BUSY_RX status. If a PD-ACK.request primitive is received while the transmitter is already busy transmitting (BUSY_TX state), the PHY entity shall discard the constructed ACK/NACK frame and issue the PD-ACK.confirm primitive with a BUSY_TX status. If the processing or transmission of PHY is not possible due to invalid parameters or for any other reason, the PHY entity shall discard the constructed ACK/NACK frame and issue the PD-ACK.confirm primitive with a FAILED status.

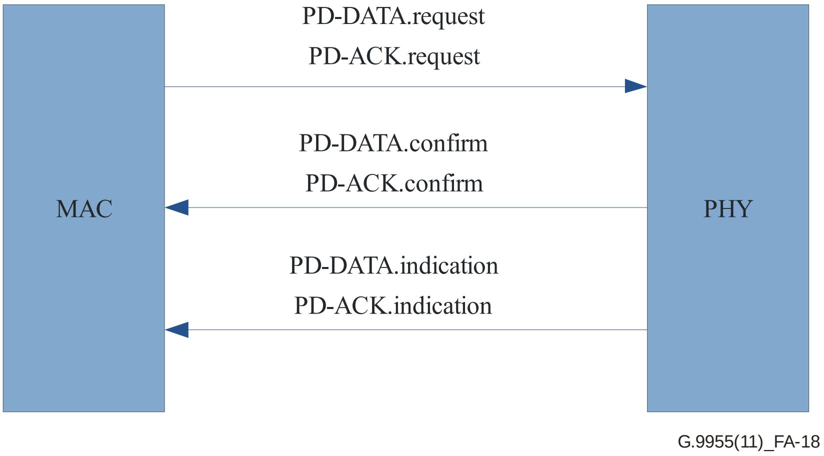


Figure 9-1 – Data or ACK primitive flow

9.1.1 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 9-1 specifies the parameters for the PD-DATA.request primitive.

Table 9-1 – The parameters for the PD-DATA.request primitive

Name	Type	Valid range	Description
psduLength	Integer	0x00-0xEF	The number of bytes contained in the PSDU to be transmitted by the PHY entity
psdu	Integer Array	Any	The set of bytes forming the PSDU request to transmit by the PHY entity

The PHY should start the transmission no later than 0.1×aSlotTime after the PD-DATA.request is issued by the MAC. The aSlotTime is defined in Table 11-13.

9.1.2 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 9-2 specifies the parameters for the PD-DATA.confirm primitive.

Table 9-2 – The parameters for the PD-DATA.confirm primitive

Name	Type	Valid range	Description
Status	Enumeration	SUCCESS, BUSY_RX, BUSY_TX, FAILED	The result of the request to transmit a packet

9.1.3 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 9-3 specifies the parameters for the PD-DATA.indication primitive.

Table 9-3 – The parameters for the PD-DATA.indication primitive

Name	Type	Valid range	Description
psduLength	Integer	0x00-0xEF	The number of bytes contained in the PSDU received by the PHY entity
psdu	Integer	–	The set of bytes forming the PSDU received by the PHY entity
ppduLinkQuality	Integer	0x00-0xFF	Link quality (LQI) value measured during reception of the PPDU

The LQI shall be measured for each received packet and is a characterization of the quality of the underlying power line channel.

The LQI is an integer ranging from 0x00 to 0xFF and LQI values in-between shall be uniformly distributed between these two limits. The LQI value is the average SNR (where averaging is done over all active tones and pilot tones, if present, in the bandplan and over all OFDM symbols in the received packet) normalized to the range from –10 dB or lower (0x00) to 53 dB or higher (0xFF), where the value of –9.75 dB is represented as 0x01 and the value of 52.75 dB is represented as

0xFE. Active tones are defined as tones which carry data (pilot tones and dummy bit tones are not included).

The LQI value is computed in the PHY and passed to the MAC with the PD-DATA.indication primitive through the ppduLinkQuality parameter – see Table 9-3. The LQI shall be measured and reported and it may be used to determine the transmission parameters, such as modulation modes.

9.1.4 PD-ACK.request

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

PD-ACK.request (

FCH

)

Table 9-4 specifies the parameter for the PD-ACK.request primitive.

Table 9-4 – The parameters for the PD-ACK.request primitive

Name	Type	Valid range	Description
FCH	Structure	Clause 7.4 PHY	The MAC layer provides all frame control header parameters described in clause 7.4 to construct FCH frame for ACK.

9.1.5 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 9-5 specifies the parameter for the PD-ACK.confirm primitive.

Table 9-5 – The parameters for the PD-ACK.confirm primitive

Name	Type	Valid range	Description
Status	Enumeration	SUCCESS BUSY_RX, BUSY_TX, FAILED	Confirm transmission of ACK frame

9.1.6 PD-ACK.indication

The PD-ACK.indication primitive indicates reception of the ACK frame from the PHY to the local MAC sublayer entity. The semantics of the PD-ACK.indication primitive is as follows:

PD-DATA.indication (

FCH

)

Table 9-6 specifies the parameter for the PD-ACK.indication primitive.

Table 9-6 – The parameters for the PD-ACK.indication primitive

Name	Type	Valid range	Description
FCH	Structure	Clause 7.4 PHY	The MAC layer receives all frame control header parameters described in clause 7.4 from PHY layer.

9.2 Management primitives

There are three types of management primitives: Get, Set and Confirm. They are used to initiate commands or retrieve data from the PHY. The PLME_SET.request function configures the PHY to an initial specific function. The PLME_GET.request is to retrieve specific parameters from the PHY and the PLME_GET.confirm reports the result of an action initiated by the MAC.

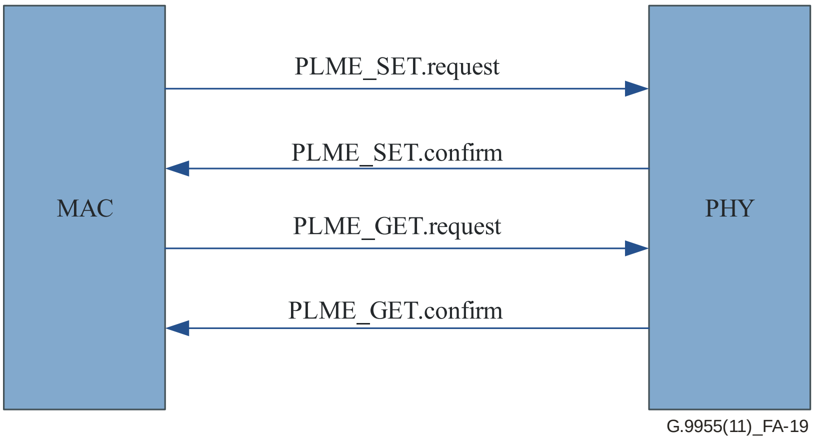


Figure 9-2 – Management primitive flow

9.2.1 PLME_SET.request

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

PLME_SET.request (

TXPower

ModulationType

ToneMap

PreEmphasis

ToneMask

DT

)

Table 9-7 specifies the parameters for the PLME_SET.request primitive.

Table 9-7 – The parameters for the PLME_SET.request primitive

Name	Type	Valid range	Description
TXPower	Integer	0x00-0x20	The MAC layer uses this primitive to notify the PHY about the gain/power setting PHY has to use to transmit the next packet.
ModulationType	Integer	0x0-0x3	This sets the TX modulation scheme for the next frame.

Table 9-7 – The parameters for the PLME_SET.request primitive

Name	Type	Valid range	Description
ToneMap	Array	0x0-0x1	Tone map parameter. The value of 0 indicates to the remote transmitter that dummy data should be transmitted on the corresponding subcarrier while a value of 1 indicates that valid data should be transmitted on the corresponding subcarrier.
PreEmphasis	Integer	0x00-0x1F	Specify transmit gain for each 10 kHz section of the available spectrum
ToneMask	Array	0x0-0x1	Tone Mask parameter. The value of 0 indicates tone is notched, 1 indicates that tone is enabled.
DT	Integer	0x00-0x07	Delimiter type as specified in Table 7-5.

9.2.2 PLME_SET.confirm

The PHY stores new parameters and returns new stored value back to the MAC layer. The semantics of the PLME_SET.confirm primitive is as follows:

PLME_SET.confirm (

TXPower

ModulationType

ToneMap

PreEmphasis

ToneMask

DT

)

Table 9-8 specifies the parameters for the PLME_SET.confirm primitive.

Table 9-8 – The parameters for the PLME_SET.confirm primitive

Name	Type	Valid range	Description
TXPower	Integer	0x00-0x20	Returns new stored value back to MAC
ModulationType	Integer	0x0-0x3	Returns new stored value back to MAC
ToneMap	Array	0x0-0x1	Returns new stored value back to MAC
PreEmphasis	Integer	0x00-0x1F	Returns new stored value back to MAC
ToneMask	Array	0x0-0x1	Returns new stored value back to MAC
DT	Integer	0x00-0x07	Delimiter type as specified in Table 7-5

9.2.3 PLME_GET.request

The PLME_GET.request primitive requests the PHY to get the parameters described in Table 9-9. The semantics of the PLME_GET.request primitive is as follows:

PLME_GET.request (

)

9.2.4 PLME_GET.confirm

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

PLME_GET.confirm (

SNR

CarrierSNR

RXSensitivity

ZCTDifferential

TXPower,

AGCGain,

ModulationType,

ToneMap,

PreEmphasis,

ToneMask,

DT

)

Table 9-9 specifies the parameters for the PLME_GET.confirm primitive.

Table 9-9 – The parameters for the PLME_GET.confirm primitive

Name	Type	Valid range	Description
SNR	Integer	0x00-0xFF	The MAC layer requests to get the channel SNR value in dB.
CarrierSNR	Integer	0x00-0x3F	The PHY provides the SNR value per each carrier.
RXSensitivity	Integer	0x0-0x1F	The PHY provides receiver sensitivity to the MAC layer.
ZCTDifferential	Integer	0x00-0xFF	The PHY computes and provides the time difference between local 50 Hz phase and remote end to the MAC layer.
TXPower	Integer	0x00-0x20	The MAC layer uses this primitive to notify the PHY about the gain/power setting PHY has to use to transmit the next packet.
AGCGain	Integer	0x0-0x3F	The MAC changes the AGC gain to a desired energy level.
ModulationType	Integer	0x0-0x3	This sets the TX modulation scheme for the next frame.
ToneMap	Array	0x0-0x1	Tone map parameter. The value of 0 indicates to the remote transmitter that dummy data should be transmitted on the corresponding subcarrier while a value of 1 indicates that valid data should be transmitted on the corresponding subcarrier.

Table 9-9 – The parameters for the PLME_GET.confirm primitive

Name	Type	Valid range	Description
PreEmphasis	Integer	0x00-0x1F	Specify transmit gain for each 10 kHz section of the available spectrum
ToneMask	Array	0x0-0x1	Tone Mask parameter. The value of 0 indicates tone is notched, 1 indicates that tone is enabled.
DT	Integer	0x00-0x07	Delimiter type as specified in Table 7-5

The SNR is an integer ranging from 0x00 to 0xFF, where values in-between shall be uniformly distributed between these two limits. The SNR values are normalized to the range from –10 dB or lower (0x00) to 53 dB or higher (0xFF), where the value of –9.75 dB is represented as 0x01 and the value of 52.75 dB is represented as 0xFE.

CarrierSNR is an integer ranging from 0x00 to 0x3F, where values in-between shall be uniformly distributed between these two limits. The CarrierSNR values are normalized to the range from –10 dB or lower (0x00) to 53 dB or higher (0x3F), where the value of –9 dB is represented as 0x01 and the value of 52 dB is represented as 0x3E.

9.2.5 PLME_SET_TRX_STATE.request

The PLME_SET_TRX_STATE.request primitive requests the PHY to change the state. The semantics of the PLME_SET_TRX_STATE.request primitive is as follows:

PLME_SET_TRX_STATE.request (

State

)

Table 9-10 specifies the parameters for the PLME_SET_TRX_STATE.request primitive.

Table 9-10 – The parameters for the PLME_SET_TRX_STATE.request primitive

Name	Type	Valid range	Description
State	Enumeration	TXON_RXOFF TXOFF_RXON	Turns off the RX PHY when transmitting packet. Turns off the transmitter and enable RX when PHY is not transmitting.

9.2.6 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 9-11 specifies the parameters for PLME_SET_TRX_STATE.confirm primitive.

Table 9-11 – The parameters for the PLME_SET_TRX_STATE.confirm primitive

Name	Type	Valid range	Description
Status	Enumeration	SUCCESS BUSY_TX BUSY_RX	Confirm RX and TX are set or provide error message if TX or RX are busy.

9.2.7 PLME_CS.request

The PLME_CS.request primitive requests the PHY to get media status using carrier sense. The semantics of the PLME_CS.request primitive is as follows:

PLME_CS.request (

)

9.2.8 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 9-12 specifies the parameters for the PLME_CS.confirm primitive.

Table 9-12 – The parameters for the PLME_CS.confirm primitive

Name	Type	Valid range	Description
Status	Enumeration	IDLE BUSY	Power line media status

10 Physical layer specification for FCC bandplans

ITU-T G.9903 devices support operation in the FCC band, as specified in Annex B of [ITU-T G.9901].

10.1 System fundamental parameters for bandplan FCC-1

Mandatory values for the OFDM control parameters for the FCC-1 bandplan are given in Table B.3 of [ITU-T G.9901]. The frequency bands used for the FCC-1 bandplan are defined in Table B.4 of [ITU-T G.9901].

As specified in Table B.4 of [ITU-T G.9901], the subcarrier spacing is 4.6875 kHz and the number of usable subcarriers is 72. DBPSK, DQPSK and D8PSK modulation schemes are supported, resulting in an up to 300 kbit/s data rate in the normal mode of operation.

The number of symbols in each PHY frame is selected based on two parameters, the required data rate and the acceptable robustness. The number of symbols, Reed-Solomon block sizes and data rate associated with 72 tones are tabulated for several values as examples in Tables 10-1 and 10-2.

Table 10-1 – RS block size for various modulations

FCC Number of symbols	Reed-Solomon blocks (bytes) D8PSK (Out/In) (Note 1)	Reed-Solomon blocks (bytes) DQPSK (Out/In) (Note 1)	Reed-Solomon blocks (bytes) DBPSK (Out/In) (Note 1)	Reed-Solomon blocks (bytes) Robust (Out/In) (Note 2)
12	(161/145)	(107/91)	(53/37)	(12/4)
20	N/A	(179/163)	(89/73)	(21/13)
28	N/A	(251/235)	(125/109)	(30/22)
NOTE 1 – Reed-Solomon with 16 bytes parity. NOTE 2 – Reed-Solomon with 8 bytes parity.				

Table 10-2 – Data rate for various modulations (excluding FCH)

FCC Number of symbols	Data rate (bit/s) D8PSK (Note 1)	Data rate (bit/s) DQPSK (Note 1)	Data rate (bit/s) DBPSK (Note 1)	Data rate (bit/s) Robust (Note 2)
12	152,899	95,957	39,015	4,217
20	N/A	138,135	61,864	11,016
28	N/A	166,469	77,213	15,584
NOTE 1 – Reed-Solomon with 16 bytes parity. NOTE 2 – Reed-Solomon with 8 bytes parity.				

The data rate can be calculated similarly to the CENELEC-A bandplan example given in clause 7. An example of how to calculate a data rate is given below using the following parameters:

• Number of FFT points	$N = 256$
• Number of subcarriers	$N_{car} = 72$
• Number of overlapped samples	$N_O = 8$
• Number of cyclic prefix samples	$N_{CP} = 30$
• Number of FCH symbols	$N_{FCH} = 12$
• Sampling frequency	$F_s = 1.2$ MHz
• Number of symbols in preamble	$N_{pre} = 9.5$

The frame control header uses 72 bits, resulting in 12 FCH symbols. This can be calculated using:

Number of FCH symbols = ceiling $((72 \times 2 \times 6)/72) = 12$

The initial phase values that should be used to generate the preamble and modulate the first symbol of the FCH are provided in Table 10-3. The bit fields for the frame control header FCH are shown in Table 10-4.

Table 10-3 – Phase vector definition for FCC-1 bandplan

c	ϕ_c	c	ϕ_c	C	ϕ_c	c	ϕ_c
				52	10(π/8)	77	8(π/8)
				53	5(π/8)	78	14(π/8)
				54	0	79	3(π/8)
				55	12(π/8)	80	9(π/8)
				56	6(π/8)	81	15(π/8)
				57	1(π/8)	82	3(π/8)
		33	2(π/8)	58	12(π/8)	83	8(π/8)
		34	(π/8)	59	6(π/8)	84	13(π/8)
		35	(π/8)	60	0	85	π/8
		36	0	61	10(π/8)	86	5(π/8)
		37	0	62	3(π/8)	87	9(π/8)
		38	15(π/8)	63	13(π/8)	88	13(π/8)
		39	14(π/8)	64	6(π/8)	89	π/8
		40	12(π/8)	65	15(π/8)	90	4(π/8)
		41	11(π/8)	66	7(π/8)	91	7(π/8)
		42	9(π/8)	67	0	92	10(π/8)
		43	7(π/8)	68	8(π/8)	93	13(π/8)
		44	4(π/8)	69	0	94	15(π/8)
		45	π/8	70	8(π/8)	95	π/8
		46	15(π/8)	71	15(π/8)	96	3(π/8)
		47	12(π/8)	72	6(π/8)	97	4(π/8)
		48	9(π/8)	73	14(π/8)	98	5(π/8)
		49	5(π/8)	74	4(π/8)	99	7(π/8)
		50	(π/8)	75	11(π/8)	100	7(π/8)
		51	14(π/8)	76	2(π/8)	101	8(π/8)
						102	9(π/8)
						103	10(π/8)
						104	10(π/8)

Table 10-4 – FCH bit fields for FCC-1 bandplan

Field	Byte	Bit Number	Bits	Definition
PDC	0	7 to 0	8	Phase detection counter
MOD	1	7 to 5	3	Modulation type
				0: ROBO
				1: DBPSK
				2: DQPSK
				3: D8PSK 4: 16-QAM 5-7: Reserved by ITU-T
Coherent Mode	1	4	1	0: differential 1: coherent mode
DT		3 to 1	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 by ITU-T
FL		0	1	PHY frame length in PHY symbols
	2	7 to 0	8	
TM[7:0]	3	7 to 0	8	TM[7:0]: Tone map
TM[15:8]	4	7 to 0	8	TM[15:8]: Tone Map
TM[23:16]	5	7 to 0	8	TM[23:16]: Tone Map
Reserved	6	7 to 0	8	Reserved by ITU-T
Reserved	7	7 to 6	2	Reserved by ITU-T
FCCS	7-8 8	5 to 0 7 to 6	6 2	Frame control check sequence (CRC8)
ConvZeros	8	5 to 0	6	Zeros for convolutional encoder

NOTE – All reserved bits in the above table are set to 0.

10.1.1 Optional FCC bandplans

In addition to the FCC-1 Bandplan, a node can optionally support the following bandplans:

- FCC-1.a Bandplan, as specified in Table B.5 of [ITU-T G.9901]
- FCC-1.b Bandplan, as specified in Table B.5 of [ITU-T G.9901]

The number of FCH symbols for the above bandplans shall be computed according to the procedure described for the main band. For example, for FCC-1.a Bandplan, the number of FCH symbol shall be ceiling ((72 × 2 × 6)/24) = 36.

The initial phase values that shall be used to generate the preamble and modulate the first symbol of FCH for the above bandplans are provided in Tables 10-4 and 10-5.

Table 10-4 – Phase vector definition for FCC-1.a bandplan

C	ϕ_c	C	ϕ_c	c	ϕ_c
33	2(π/8)	41	12(π/8)	49	9(π/8)
34	1(π/8)	42	6(π/8)	50	14(π/8)
35	0(π/8)	43	15(π/8)	51	1(π/8)
36	14(π/8)	44	8(π/8)	52	4(π/8)
37	12(π/8)	45	0(π/8)	53	6(π/8)
38	10(π/8)	46	7(π/8)	54	8(π/8)
39	6(π/8)	47	14(π/8)	55	9(π/8)
40	1(π/8)	48	4(π/8)	56	10(π/8)

Table 10-5 – Phase vector definition for FCC-1.b bandplan

c	ϕ_c	c	ϕ_c	c	ϕ_c
65	2(π/8)	79	10(π/8)	93	1(π/8)
66	1(π/8)	80	4(π/8)	94	5(π/8)
67	1(π/8)	81	14(π/8)	95	9(π/8)
68	0(π/8)	82	7(π/8)	96	13(π/8)
69	14(π/8)	83	0(π/8)	97	0(π/8)
70	13(π/8)	84	8(π/8)	98	3(π/8)
71	11(π/8)	85	0(π/8)	99	5(π/8)
72	8(π/8)	86	7(π/8)	100	6(π/8)
73	5(π/8)	87	15(π/8)	101	7(π/8)
74	1(π/8)	88	5(π/8)	102	9(π/8)
75	14(π/8)	89	12(π/8)	103	9(π/8)
76	9(π/8)	90	2(π/8)	104	10(π/8)
77	4(π/8)	91	7(π/8)		
78	15(π/8)	92	13(π/8)		

10.1.2 Optional coherent mode

This clause describes the operation of ITU-T G.9903 devices in the FCC bandplan when operating in the optional coherent mode. This clause only describes the portions of the standard that are different from the main differential mode. Portions of the coherent transmitter that are not described here shall operate exactly as described in the differential mode.

10.1.2.1 Frame structure

In a similar way to differential mode, the coherent mode shall support two types of frames: data frames and ACK/NACK frames. The frame structure of data frames shall be identical to the one used in differential mode except for two changes:

a) The data portion of the PHY frame shall be preceded by an S1 symbol followed by an S2 symbol, where both symbols shall be inserted between the last FCH symbol and the first data symbol. The S2 symbol shall have the same phase reference vector used in differential mode for a P symbol. The only difference from a P symbol is that the S2 symbol consists of a P symbol plus a cyclic prefix of 30 samples and an overlap of 8 samples, resulting in 278 samples when an IFFT size of 256 is used. Hence, the duration of the S2 symbol shall be the same as for that of an FCH symbol or a data symbol. The S1 symbol shall be an inverted S2 symbol (i.e., –S2), hence it will also consist of 278 samples.

b) Pilot tones shall be inserted in the data symbols as described in clause 10.1.2.14 on pilot tones.

c) The FCH shall be coherently modulated.

The frame structure of the ACK/NACK frames for coherent mode shall be identical to the one used in differential mode.

10.1.2.2 Preamble

The preamble for coherent mode is composed of 8 or (8+4=12) identical P symbols followed by an M symbol that is followed by a half M symbol. The P and M symbols for coherent mode are identical to the ones generated in differential mode. Hence, the only difference between the preamble sequence for coherent and differential modes is that for coherent mode one S1 followed by one S2 symbols are inserted between the last FCH symbol and the first data symbol. The initial phases used for both modes are shown in Table 10-3.

All coherent mode preamble symbols (P and M and the additional symbols between the last FCH symbol and the first data symbol) shall have the same gain factor compared to data symbols. The gain is defined to be 3 dB.

10.1.2.3 Frame control header

The twelve symbols immediately after the preamble are reserved for a frame control header (FCH) whose format is identical to the one generated in differential mode. The "Coherent Mode" bit in the FCH shall be used to indicate whether the payload is modulated differentially or coherently. The frame control header itself shall be modulated coherently.

10.1.2.4 CRC8

An 8-bit cyclic redundancy check (CRC) is used for error detection in the FCH. The CRC8 is computed as a function of the 58-bit sequence using an initial value of 0xFF. The CRC8 is calculated using the following eight degree generator polynomial:

$$G(x) = x^8 + x^2 + x + 1$$

Data bits are shifted to the CRC8 register starting with the most significant bit of the first byte of the FCH. The CRC8 is the remainder of the division of the FCH polynomial by the generator polynomial. The ones complement of the remainder is transmitted starting with the highest order bits and ending with the lowest order bit.

10.1.2.5 Data scrambler

The data scrambler used in coherent mode shall be identical to the one used in differential mode.

10.1.2.6 FEC coding

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

The FEC encoder for coherent mode shall be identical to the one used for differential mode. In particular, Reed-Solomon encoding, convolutional encoding and repetition coding by 4 and 6 shall all be identical to differential mode.

10.1.2.7 Payload padding

The encoded output (both FCH and payload) shall be padded to fit the encoded bits to an integer number of OFDM symbols. The padding is done by appending '0's at the end to fit the encoded bits into an integer number of OFDM symbols.

10.1.2.8 Interleaver

The interleaver for coherent mode shall be identical to the interleaver in differential mode where the pilot tones shall not be considered part of the active tones and hence shall be completely ignored by the interleaver. This means that the number of subcarriers 'm' shall not include in it the pilot tones (nor the masked tones as is the case for differential mode).

10.1.2.9 Coherent mapping for BPSK, QPSK, 8PSK, 16QAM and robust modes

The mapping block is responsible for assuring that the transmitted signal conforms to the given tone map and tone mask. The tone map and mask are concepts of 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.

Data bits are mapped for coherent modulation (BPSK, QPSK, 8PSK, 16QAM or robust) as follows: For a given symbol, instead of using the same carrier, the previous symbol as its phase reference, it uses the preamble phase of the same carrier as its reference. This predefined phase reference is identical to the one that is specified for differential modulation as shown in Table 10-3. Both the FCH symbols and data symbols use the same phase reference vector.

10.1.2.10 Mapping for BPSK and robust modulations

In BPSK (and robust) modulation a phase shift of 0° represents a binary "0" and a phase shift of 180° represent a binary "1" as illustrated in Table 10-6.

Table 10-6 – BPSK and robust encoding table of k-th subcarrier

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

The constellation shall be identical to the one used for differential mode.

10.1.2.11 Mapping for QPSK modulation

In QPSK a pair of 2 bits is mapped to 4 different output phases. The phase shifts of 0°, 90°, 180° and 270° represent binary "00", "01", "11" and "10", respectively, as illustrated in Table 10-7.

Table 10-7 – QPSK encoding table of k-th subcarrier

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

The constellation shall be identical to the one used for differential mode.

10.1.2.12 Mapping for 8PSK modulation

In 8PSK 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° represent binary 000, 001, 011, 010, 110, 111, 101 and 100 respectively, as illustrated in Table 10-8.

Table 10-8 – 8PSK encoding table of kth subcarrier

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

The constellation shall be identical to the one used for differential mode.

10.1.2.13 Mapping for 16QAM modulation

In 16-QAM modulation, 4 bits are mapped to one of sixteen different constellation points. The mapping is shown in Figure 10-1 and Table 10-9.

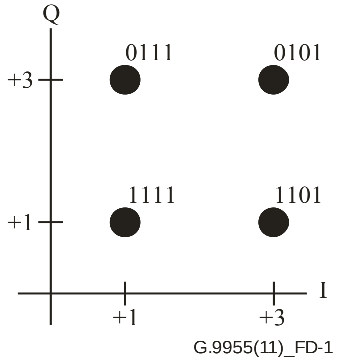


Figure 10-1 – 16-QAM constellation diagram

The complete constellation description is given in Table 10-9.

Table 10-9 – Mapping for 16QAM

Bits [d_1d_0]	*I*
00	–3
10	–1
11	1
01	3

Bit [d_3d_2]	*Q*
00	–3
10	–1
11	1
01	3

10.1.2.14 Pilot tones

Pilot tones can be used in coherent mode to help with clock recovery and channel estimation, particularly in harsh environments where strong noise and frequent channel variations occur.

For pilot assignment, the pilot indices shall be sequentially enumerated over only the active subcarrier set:

$$P(i,j) = (OFFSET + (FreqSpacing \times i) + 2 \times j)\%M_{ACTIVE} \qquad (10\text{-}1)$$

Where:

– P(i,j) is the relative position of pilot i in symbol j within the set of active subcarriers. The set of active subcarriers is enumerated as 0, 1, 2….., M_{ACTIVE}-1

– M is the number of subcarriers per symbol in a given band [FCC-1: M= 72; FCC-1.a: M=24; FCC-1.b: M=40]

– M_{ACTIVE} is the number of active subcarriers ($M_{ACTIVE} \leq M$)

– FreqSpacing = Frequency spacing between pilots in same symbol [12 for all FCC bands]

– i = pilot index = 0,1,2,…,ceil(M_{ACTIVE} /FreqSpacing)-1

– j = symbol number = 0,1,2,3,…, N-1

– N = total number of data symbols per frame.

– OFFSET = X [FCC-1:X = 36; FCC-1.a: X = 0; FCC-1.b: X = 0]

The absolute pilot tone index with respect to FFT numerology is given by:

$$Pabs(i,j) = STARTINDEX+ Q_{ACTIVE} (P(i,j)) \qquad (10\text{-}2)$$

Where:

– Q= [0, 1, 2,..., M-1] is a vector of the relative indices of the subcarriers of a given band. Length(Q) = M

– Q_{ACTIVE} is a vector of the relative indices of the active subcarriers of the given band. Q_{ACTIVE} is derived from Q by removing the non-active (i.e., masked) subcarriers. Length(Q_{ACTIVE}) = M_{ACTIVE}

– STARTINDEX corresponds to the first subcarrier in the band plan:

STARTINDEX = Y [FCC-1 Y= 33;
FCC-1.a Y=33;
FCC-1.b Y=65]

The pilot tones shall consist of sine waves at the specified tone frequencies modulated in QPSK using the constellation specified. The bits that get mapped to the constellation points shall be generated from a pseudo-random sequence using a linear feedback shift register (LFSR) with the following polynomial:

$$p(x) = x^7 + x^4 + 1$$

as shown in Figure 10-2.

The bits in the LFSR shall be initialized to all-ones at the start of each PHY frame.

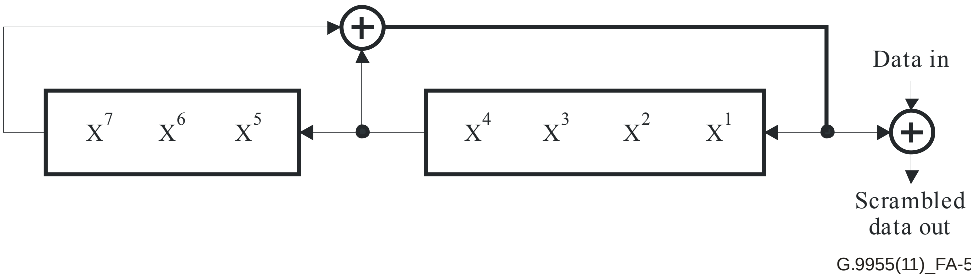


Figure 10-2 – LFSR used to generate the data bits that are used to modulate the pilot tones

The LFSR shall only generate bits used for modulating pilots. For every two consecutive output bits from the LFSR, the first bit shall be mapped to the LSB of the QPSK symbol and the second bit shall be mapped to the MSB of the QPSK symbol.

10.1.2.15 Frequency domain pre-emphasis

For further study.

10.1.2.16 OFDM generation (IFFT and CP addition)

OFDM generation for coherent mode shall be identical to the procedure used for differential mode; see clause 7-10.

10.1.2.17 Windowing

Windowing for coherent mode shall follow the identical procedure used for differential mode; see clause 7-11.

10.1.3 Error vector magnitude limits

For the EVM calculation, the procedure given in clause 8.5.2 can be used here with the following changes:

1) The number of subcarriers is 72 instead of 36.

2) For the preamble EVM calculation the 6 symbols should be used starting from the third symbol:

a) $\text{Tot\_En}^{(ref)} = \sum_{i=2}^{7} \text{Avg\_En}_i^{(ref)}$

b) $Total\_MSE = \sum_{i=2}^{7} MSE_i$

The values of EVM calculated for both data and preamble symbols shall not exceed the values given in Table 10-10.

Table 10-10 – Maximum allowed EVM values

Modulation	EVM, dB (Note)
1, 2, 3 bits	–15
4 bits	–19
NOTE – These EVM requirements shall be met for all applied transmit power levels; however, for 3 and 4 bit modulations, the transmit power levels under which these requirements are met may be lower than those for 1 and 2 bit modulation.	

11 Data link layer specifications

11.1 Introduction

The ITU-T G.9903 data link layer specification comprises two sublayers:

- the MAC sublayer based on [IEEE 802.15.4] and
- the adaptation sublayer based on [IETF RFC 4944].

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

11.2 Conventions

In the present clause, 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 shall apply without modifications or remarks.

– S = "Selection": The statements of the reference document shall apply with the selections specified.

– E = "Extension": The statements of the reference document shall apply with the extensions (modifications and remarks noted under the part title) specified.

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

11.3 MAC sublayer specification

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

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

The MAC services and primitives are as given in clauses 7.1.1 to 7.1.17 of [IEEE 802.15.4] together with the following statements and modifications shown in Table 11-1.

References to clauses in the "Clause" column refer to the referenced document, while references to clauses/annexes in the "Title and remarks" column refer to this Recommendation unless specifically indicated otherwise. The interpretation of the statement column is given in clause 11.2.

Table 11-1 – Selections from IEEE 802.15.4 clause 7.1

Clause	Title and 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 clause 11.3.1.2. – Only non-beacon-enabled PAN is used – Bit b2 of TxOptions parameter shall always be 0 See Annex D for the 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 – Modification: Time stamp is optional and defined as the absolute time in milliseconds at which the frame was created and eventually after the encryption (32 bit value).	S, E
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 11.3.1.2. – Modification: Time stamp is optional and defined as the absolute time in milliseconds at which the frame was received and constructed, decrypted (assuming encryption was valid) (32 bit value). See Annex D for the 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	N/R
7.1.1.5	MCPS-PURGE.confirm	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 11.4.5.	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 11.4.5.	N/R

Table 11-1 – Selections from IEEE 802.15.4 clause 7.1

Clause	Title and remarks/modifications	Statement
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 11.4.5.	N/R
7.1.3.4	MLME-ASSOCIATE.confirm – MLME-ASSOCIATE.confirm is not used in this specification. Association is performed by the 6LoWPAN bootstrap protocol described in clause 11.4.5.	N/R
7.1.3.5	Association message sequence chart – The association message sequence chart described in Figure 31 shall be ignored for this specification, as association is performed using the bootstrap mechanism described in clause 11.4.5.	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 11.4.5.	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 11.4.5.	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 11.4.5.	N/R
7.1.4.4	Disassociation message sequence chart – The disassociation message sequence chart described in Figure 31 shall be ignored for this specification, as disassociation is performed using the bootstrap mechanism described in clause 11.4.5.	N/R
7.1.5	Beacon notification primitive	N
7.1.5.1	MLME-BEACON-NOTIFY.indication – Only nonbeacon-enabled PANs are used. – This primitive is generated upon receipt of a beacon during an active scan.	S
7.1.5.1.1	Semantics of the service primitive MLME-BEACON-NOTIFY.indication (PANDescriptor) PANDescriptor is described in Table F.6.	S
7.1.5.1.2	When generated – This primitive is generated upon receipt of a beacon during an active scan.	S
7.1.5.1.3	Appropriate usage	N
7.1.6	Primitives for reading PIB attributes	N
7.1.6.1	MLME-GET.request	N

Table 11-1 – Selections from IEEE 802.15.4 clause 7.1

Clause	Title and remarks/modifications	Statement
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
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 shall be set to 0. – The ChannelPage parameter is not used and shall be set to 0. – The SecurityLevel shall 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 shall 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

Table 11-1 – Selections from IEEE 802.15.4 clause 7.1

Clause	Title and remarks/modifications	Statement
7.1.11.2	MLME-SCAN.confirm During active scan, MLME-BEACON-NOTIFY.indication is generated in response to MLME-SCAN.request as soon as a beacon is received.	N
7.1.11.2.1	Semantics of the service primitive	S
7.1.11.2.2	When generated	S
7.1.11.2.3	Appropriate usage	N
7.1.11.3	Channel scan message sequence chart – Figure 79 shall be ignored (ED scan not supported) – Figure 82 shall be ignored (passive scan not supported) – Figure 86 shall be ignored (orphan scan not supported) – Active scan message sequence chart is specified in clause 11.4.5.2.2 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
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 11.5.1).	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 shall be set as described in clause 11.5.1.	S

Table 11-1 – Selections from IEEE 802.15.4 clause 7.1

Clause	Title and remarks/modifications	Statement
7.1.14.1.2	Appropriate usage	N
7.1.14.1.3	Effect on receipt – Primitive parameters shall be set as described in clause 11.5.1.	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 shall 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 11.5.2.	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
NOTE – Time stamp shall refer to a free running counter in milliseconds. The counter is initialized to zero at node start-up.		

11.3.1.2 Extensions to IEEE 802.15.4 clause 7.1: additional QualityOfService parameter

As shown in Table 11-2, the quality of service (QOS) parameter defines the level of priority assigned to the MSDU to be transmitted. Annex C defines the priority mechanism of ITU-T G.9903 devices.

Table 11-2 – 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

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

11.3.2.1 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] apply, with the selections specified in Table 11-3.

Table 11-3 – Selections from clause 7.2 of [IEEE 802.15.4]

Clause	Title and remarks/modifications	Statement
7.2	MAC frame formats	N
7.2.1	General MAC frame format – Segment control fields are added to the MHR (see clause 11.3.2.2) – Detailed descriptions of the segment control fields are shown in Table 11-5.	E
7.2.1.1	Frame control field NOTE – The Ack request field must be set with a value consistent with the PHY layer DT field.	N
7.2.1.1.1	Frame type subfield – The present specification does not use an acknowledgement frame type value. – The detailed ACK implementation is described in Annex E. An acknowledgement can be sent by invoking the PD-ACK.request primitive.	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 shall be set to 0.	S
7.2.1.1.4	Acknowledgement request subfield – The present specification translates the acknowledgement request subfield to the proper delimiter type of frame control header. – The detailed ACK implementation is described in Annex E. An acknowledgement can be sent by invoking the PD-ACK.request primitive.	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 shall 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 6 bytes (see clause 12)	S
7.2.1.8	Frame payload field	N
7.2.1.9	FCS field	N

Table 11-3 – Selections from clause 7.2 of [IEEE 802.15.4]

Clause	Title and remarks/modifications	Statement
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 shall 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 shall 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 shall 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 shall be set to 0. – Within the framework of the present Recommendation, 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 shall be set to 1. If another profile is used, this field shall 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 shall be set to 0 (GTS are not supported). – The PAN coordinator never accepts a GTS request, therefore the GTS permit parameter of the GTS specification field shall be set to 0.	S
7.2.2.1.4	GTS direction field – The GTS feature is not used and the GTS direction field shall 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 [IEEE 802.15.4], this list shall 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 the '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 shall not be present in beacons.	N/R

Table 11-3 – Selections from clause 7.2 of [IEEE 802.15.4]

Clause	Title and remarks/modifications	Statement
7.2.2.1.8	Beacon payload field – The beacon payload field is comprised of a one byte estimate of the route cost to the coordinator (RC_COORD). The route cost shall be based on the route cost calculation in LOAD. RC_COORD may be approximated by saving the lowest route cost extracted from the PREP, RREQ or RREP packets that originated from the PAN coordinator. If a device has failed to communicate with the PAN coordinator it shall set RC_COORD to its maximum value of 0xFF. A device shall initialize RC_COORD to 0x7F on association. The PAN coordinator shall set its RC_COORD to 0x00.	S, E
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 – The 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 Annex E. An acknowledgement can be sent by invoking the PD-ACK.request primitive.	S
7.2.2.4	MAC command frame format	N
7.2.2.4.1	MAC command frame MHR fields	N
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

11.3.2.2 Extensions to IEEE 802.15.4 clause 7.2: MAC frame formats

Tables 11-4 and 11-5 define the segment control field added in the MAC header (MHR) specified in [IEEE 802.15.4], clause 7.2.

Table 11-4 – General MAC frame format

Octets: 3	2	1	0/2	0/2/8	0/2	0/2/8	0/6	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	MF R

Table 11-5 – Segment control fields

Field	Byte	Bit number	Bits	Definition
RES	0	7-4	4	Reserved by ITU-T
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

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

11.3.3.1 Selections from IEEE 802.15.4 clause 7.3: MAC command frames

The MAC frame formats described in clause 7.3 of [IEEE 802.15.4] apply, with the selections specified in Table 11-6.

Table 11-6 – Selections from clause 7.3 of [IEEE 802.15.4]

Clause	Title and remarks/modifications	Statement
7.3	MAC command frames – All devices are Full Function Devices – The supported command list is defined in 9.3.3.2.1	S, E
7.3.1	Association request command – Within the framework of the present Recommendation, association is performed by the 6LoWPAN Bootstrap protocol described in clause 11.4.5.2.2, so 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 Recommendation, association is performed by the 6LoWPAN Bootstrap protocol described in clause 11.4.5.2.2, so 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 Recommendation, association is performed by the 6LoWPAN Bootstrap protocol described in clause 11.4.5.2.2, so clause 7.3.2 of [IEEE 802.15.4] is not relevant.	N/R
7.3.4	Data Request command	N/R

Table 11-6 – Selections from clause 7.3 of [IEEE 802.15.4]

Clause	Title and remarks/modifications	Statement
7.3.5	PAN ID conflict notification command – PAN ID conflict notification is performed by the adaptation layer, see clause 11.5.2.	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 shall 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

11.3.3.2 Extensions to IEEE 802.15.4 clause 7.3: MAC command frames

11.3.3.2.1 MAC command frames supported

The present Recommendation supports the MAC command frames described in Table 11-7.

Table 11-7 – MAC command frames

Command frame identifier	Command name	Clause
0x00-0x06	Reserved by ITU-T	–
0x07	Beacon request	See clause 7.3.7 of [IEEE 802.15.4]
0x08-0x09	Reserved by ITU-T	–
0x0A	Tone map response	See clause 11.3.3.2.2
0x0B-0xFF	Reserved by ITU-T	–

11.3.3.2.2 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 response contains the number of used tones and allocation (tone map), modulation mode and TX power control parameters. The tone map response command frame shall be formatted as illustrated in Table 11-8.

The channel estimation response command frame shall be formatted as illustrated in Table 11-8:

Table 11-8 – Tone map response format

Octets: **(see clause 7.2.2.4 of [IEEE 802.15.4])**	**1**	**7 (for CENELEC-A band)** **15 (for FCC band)**	**2**
MHR fields	Command frame identifier (see Table 11-9)	Tone map response payload (see Table 11-9)	MFR Fields

The tone map response message parameters are shown in Table 11-9 for the case of CENELEC-A and in Table 11-10 for the case of FCC.

Table 11-9 – 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 mode 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 the number of gain steps requested for the tones represented by TM[0] (optional)
TXCOEF[7:4]	3	3-0	4	Specifies the number of gain steps requested for the tones represented by TM[1] (optional)
TXCOEF[11:8]	4	7-4	4	Specifies the number of gain steps requested for the tones represented by TM[2] (optional)
TXCOEF[15:12]	4	3-0	4	Specifies the number of gain steps requested for the tones represented by TM[3] (optional)
TXCOEF[19:16]	5	7-4	4	Specifies the number of gain steps requested for the tones represented by TM[4] (optional)
TXCOEF[23:20]	5	3-0	4	Specifies the number of gain steps requested for the tones represented by TM[5] (optional)
Reserved by ITU-T	6	7-0	8	Shall be set to zero
NOTE – As also mentioned in clause 7.4, TM[8] to TM[6] are set to zero and are not used in CENELEC-A band.				

Table 11-10 – Tone map response message description for FCC 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	2-0	3	Modulation type: 0 – Robust mode 1 – DBPSK 2 – DQPSK 3 – D8PSK 4 – 16-QAM (Note) 5-7: reserved by ITU-T
TM[0:7]	1	7-0	8	Tone map [0:7]
TM[8:15]	2	7-0	8	Tone map [8:15]
TM[16:23]	3	7-0	8	Tone map [16:23]
LQI	4	7-0	8	Link quality indicator
TXCOEF[1:0]	5	7-6	2	Specifies the number of gain steps requested for the tones represented by TM[0] (optional)
TXCOEF[3:2]	6	5-4	2	Specifies the number of gain steps requested for the tones represented by TM[1] (optional)
TXCOEF[5:4]	6	3-2	2	Specifies the number of gain steps requested for the tones represented by TM[2] (optional)
TXCOEF[7:6]	6	1-0	2	Specifies the number of gain steps requested for the tones represented by TM[3] (optional)
….	…	…	…	…
TXCOEF[47:46]	10	1-0	2	Specifies the number of gain steps requested for the tones represented by TM[23] (optional)
Reserved	11	8	8	Reserved by ITU-T
NOTE – The coherent mode specified in clause 10.1.2 is optional.				

Where:

- MOD: a 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 four modulation modes (DBPSK, DQPSK, D8PSK or robust mode) it wants the transmitter to use when sending the next data frame. Table 11-11 (a and b) lists the allowed bit values and the modulation modes they correspond to.

Table 11-11a – Modulation method field for CENELEC-A band

MOD value	Interpretation
00	Robust modulation
01	DBPSK modulation
10	DQPSK modulation
11	D8PSK modulation

Table 11-11b – Modulation method field for FCC band

MOD value	Interpretation
000	Robust modulation
001	DBPSK modulation
010	DQPSK modulation
011	D8PSK modulation
100-111	Reserved by ITU-T

- TXRES: a parameter that specifies the transmit gain resolution corresponding to one gain step.
- TXGAIN: a parameter that specifies to the transmitter the total amount of gain that it shall 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 4-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 the previous frame (default value).
- TM: a 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 shall be transmitted on the corresponding subcarrier while a value of "1" indicates that valid data shall be transmitted on the corresponding subcarrier.
- TXCOEF (optional): a parameter that specifies transmitter gain for each group of tones represented by one valid bit of the tone map. 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 group of tones is mapped to a 4-bit value for CENELEC-A or a 2-bit value for FCC where a "0" in the most significant bit indicates a positive gain value, hence an increase in the transmitter gain scaled by TXRES is requested for that section and a "1" indicates a negative gain value, hence a decrease in the transmitter gain scaled by TXRES 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 shall be used.
- The LQI value is computed in the PHY and passed to the MAC with the PD-DATA.indication primitive through the ppduLinkQuality parameter – see Table 9-3.

On receipt 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 may be added, based on implementation-dependent limitations. The neighbour table is defined in Table 11-17.

The following procedure shall 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 does not exist or has not aged (age counter is 0), the MAC sublayer sets the TMR bit of an outgoing packet segment control field and requests new tone map information. In this case the MAC data shall be sent in robust mode:

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

   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 shall measure the per-carrier quality of the channel, construct and send a tone map response message back to the originator station. The destination station shall 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.

   The destination station shall attempt to send a tone map response message as soon as possible after receiving a tone map request message from the source station.

   If the source station receives a tone map response message, it will update a neighbour table record related to the destination address with a new tone map, modulation and TX gain parameters. If the record does not exist, the MAC sublayer will create a new one. The age counter shall be set to a defined value (see macMaxAgeTime in clause 11.3.4.2.2). 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 reaches the value "0".

   If the source station does not receive a tone map response message after transmitting a tone map request message to a certain destination, it shall 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.

   The MAC sublayer shall not send a tone map request message to the destination device if no data has been sent to this device.

The tone map request/response message sequence chart is shown in clause 11.3.7.2.4.

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

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

The MAC frame formats described in clause 7.4 of [IEEE 802.15.4] apply, with the selections specified in Table 11-12.

Table 11-12 – Selections from clause 7.4 of [IEEE 802.15.4]

Clause	Title and remarks/modifications	Statement
7.4	MAC constants and PIB attributes	N
7.4.1	MAC constants – The aBaseSlotDuration parameter is not used and shall be set to 0. – The aBaseSuperframeDuration parameter is not used and shall be set to 0. – The aExtendedAddress parameter shall be equal to the EUI-48 address of the device mapped to an EUI-64 address. – The aGTSDescPersistenceTime parameter is not used and shall be set to 0. – The aMaxBeaconOverhead parameter shall be set to 0. – The aMaxBeaconPayloadLength parameter is not used and shall be set to. – The aMaxLostBeacons parameter is not used and shall be set to 0. – The aMaxMACSafePayloadSize parameter is not used and shall be set to 0. – The aMaxMACPayloadSize parameter is fixed to 400 bytes by the present Recommendation. – The aMaxMPDUUnsecuredOverhead parameter is not used and shall be set to 0. – The aMaxSIFSFrameSize parameter is not used and shall be set to 0. – The aMinCAPLength parameter is not used and shall be set to 0. – The aMinMPDUOverhead parameter is not used and shall be set to 0. – The aNumSuperframeSlots parameter is not used and shall be set to 0. – The aUnitBackoffPeriod parameter shall be set to aSlotTime. – Extensions: Additional MAC sublayer constants are defined in clause 11.3.4.2.1.	S, E
7.4.2	MAC PIB attributes – The macAckWaitDuration parameter shall be set according to the following formula: macAckWaitDuration = aRIFS + aAckTime + aCIFS – The macAssociatedPANCoord parameter is not used and shall be set to FALSE. – The macAssociationPermit parameter is not used and shall be set to TRUE. – The macAutoRequest parameter is not used and shall be set to FALSE. – The macBattLifeExt parameter is not used; changing it has no effect on the behaviour of the device. Its default value shall be FALSE. – The macBattLifeExtPeriods parameter is not used; changing it has no effect on the behaviour of the device. Its default value shall be 0. – The macBeaconPayload parameter is not used; changing it has no effect on the behaviour of the device. Its default value shall be NULL. – The macBeaconPayloadLength parameter is not used and shall be set to 0. – The macBeaconOrder parameter is not used; changing it has no effect on the behaviour of the device. Its default value shall be left to 15. – When the macBeaconTxTime parameter reaches 0xFFFFFF, it shall not change anymore.	

Table 11-12 – Selections from clause 7.4 of [IEEE 802.15.4]

Clause	Title and remarks/modifications	Statement
	– The macGTSPermit parameter is not used; changing it has no effect on the behaviour of the device. Its default value shall be FALSE. – The macMaxBE parameter is fixed to 5 by the present Recommendation. – The macMaxCSMABackoffs default value is fixed to 8 by the present Recommendation. – The macMaxFrameTotalWaitTime parameter is not used and shall be set to 0. – The macMinBE parameter is fixed to 3 by the present Recommendation. – 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 shall be set to macAckWaitDuration. – The macRxOnWhenIdle parameter shall be set to TRUE. – The macSecurityEnabled parameter shall be set to TRUE. – The macShortAddress parameter shall 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 shall be left to 15. – The macSyncSymbolOffset is not used and shall be set to 0. – The macTimestampSupported parameter shall be set to TRUE. – The macTransactionPersistenceTime parameter is not used and shall be set to 0. – Extensions: Additional set of IB attributes are defined in clause 11.3.4.2.	S, E

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

11.3.4.2.1 Additional MAC sublayer constants to IEEE 802.15.4 clause 7.4.1

Table 11-13 defines the list of MAC sublayer constants added by the present Recommendation.

Table 11-13 – Additional MAC sublayer constants to clause 7.4.1 of [IEEE 802.15.4]

Constant	Description	Value
aPreamSymbolTime	Defines the duration of one preamble symbol on the physical layer (in microseconds).	640
aSymbolTime	Defines the duration of one data symbol on the physical layer (in microseconds).	695
aSlotTime	The duration of the contention slot time (in data symbols)	2
aCIFS	Defines the contention interframe space (number of data symbols). It is defined in Annex C.	8 for CENELEC-A 10 for FCC

Table 11-13 – Additional MAC sublayer constants to clause 7.4.1 of [IEEE 802.15.4]

Constant	Description	Value
aRIFS	Defines the response inter-frame space (number of data symbols). It is defined in Annex C.	8 for CENELEC-A 10 for FCC
aEIFS	Defines the duration of the extended interframe space. It is defined in Annex C.	aSymbolTime × (aMaxFrameSize + aRIFS + aCIFS) + aAckTime
aMinFrameSize	Defines the minimum MAC frame size in data symbols.	4
aMaxFrameSize	Defines the maximum MAC frame size in data symbols.	252
aAckTime	Defines the duration of acknowledgement: N_{PRE} – number of preamble symbols is defined in clause 7.1 N_{FCH} – number of FCH symbols is defined in clause 7.1	N_{PRE} × aPremSymbolTime + N_{FCH} × aSymbolTime

11.3.4.2.2 Additional MAC sublayer attributes to IEEE 802.15.4 clause 7.4.2

Table 11-14 defines the list of MAC sublayer attributes added by the present Recommendation.

Table 11-14 – Additional attributes to clause 7.4.2 of [IEEE 802.15.4]

Attribute	Identifier	Type	Range	Description	Default value
macHighPriorityWindowSize	0x01000113	Unsigned integer	1-7	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	0x02000102	Unsigned integer	0-4 294 967 295	Statistic counter of successfully received MSDUs	0
macTxCmdPacketCount	0x02000201	Unsigned integer	0-4 294 967295	Statistic counter of successfully transmitted command packets	0
macRxCmdPacketCount	0x02000202	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

Table 11-14 – Additional attributes to clause 7.4.2 of [IEEE 802.15.4]

Attribute	Identifier	Type	Range	Description	Default value
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
macNeighbourTable	0x0000006B	Set	–	The neighbour table defined in clause 11.3.5.2.	–
macNumberOfHops	0x0000006C	Unsigned integer	0-14	The number of hops to reach the PAN coordinator.	8
macFreqNotching	0x00006D	Bool	FALSE TRUE	S-FSK 63 and 74 kHz frequency notching. Default value is FALSE (disabled)	FALSE
macCSMAFairnessLimit	0x02000112	Unsigned integer	0-255	Specifies how many failed back-off attempts, back-off exponent is set to minBE	15

Table 11-14 – Additional attributes to clause 7.4.2 of [IEEE 802.15.4]

Attribute	Identifier	Type	Range	Description	Default value
macMaxAgeTime	0x02000113	Unsigned integer	0-255	Maximum lifetime of a device in neighbour table in minutes before sending a new tone map request	2
macMaxNeighborValidTime	0x02000114	Unsigned integer	0-255	Maximum time of validity for an entry in neighbour table in minutes	255
macRCCoord	0x02000115	Unsigned integer	0-255	Route cost to coordinator to be used in the beacon payload as RC_COORD	255

11.3.4.2.3 MAC sublayer attributes and their associated ID

Table 11-15 defines the new identifier associated with IEEE 802.15.4 MAC sublayer attributes used by the present Recommendation:

Table 11-15 – MAC sublayer attributes and their associated ID

Attribute	Identifier[1]	Type	Range	Description	Default value
macAckWait Duration	0x01000103	Integer	0x0-0xFFFF	Duration of acknowledgement in microseconds	aSymbolTime ×(aRIFS + aCIFS)+ aAckTime
macBSN	0x01000105	Integer	0x0-0xFF	Beacon frame sequence number	random
macCoordExtended Address	0x01000106	Set of 8 bytes	–	Coordinator extended address	0x0
macCoordShort Address	0x01000107	Integer	0x0-0xFFFF	Coordinator short address	0x0
macDSN	0x01000108	Integer	0x0-0xFF	Data frame sequence number	random
macMaxBE	0x0100010A	Integer	0-20	Maximum value of back-off exponent. It should always be > macMinBE	8

Table 11-15 – MAC sublayer attributes and their associated ID

Attribute	Identifier[1]	Type	Range	Description	Default value
macMaxCSMA Backoffs	0x0100010B	Integer	0-0xFF	Maximum number of back-off attempts	50
macMaxFrame Retries	0x0100010D	Integer	0-10	Maximum number of retransmission	5
macMinBE	0x0100010E	Integer	0-20	Minimum value of back-off exponent	3
macPanId	0x0100010F	Integer	0x0-0xFFFF	PAN Id	0xFFFF
macResponseWait Time	0x01000110	Integer	0x0-0xFFFF	Response waiting time in microseconds, set to macAckWait Duration	aSymbolTime ×(aRIFS + aCIFS)+ aAckTime
macSecurityEnabled	0x01000111	Boolean	–	Security enabled	TRUE
macShortAddress	0x01000112	Integer	0x0-0xFFFF	Device short address	0xFFFF
macPromiscuous Mode	0x01000115	Boolean	–	Promiscuous mode enabled	FALSE
macTimeStamp Support	0x0000005C	Boolean	–	MAC frame time stamp support enable	TRUE

[1] These are new identifiers associated with IEEE 802.15.5 MAC sublayer attributes that are used by this Recommendation.

11.3.5 MAC functional description (based on clause 7.5 of IEEE 802.15.4)

11.3.5.1 Selections from clause 7.5 of IEEE 802.15.4: MAC functional description

The MAC functional description described in clause 7.5 of [IEEE 802.15.4] applies, with the selections specified in Table 11-16.

Table 11-16 – Selections from clause 7.5 of [IEEE 802.15.4]

Clause	Title and remarks/modifications	Statement
7.5	MAC functional description – beacon-enabled PAN and GTS are not supported – GTS contention free access is not supported	S
7.5.1	Channel access – See Annex C for the channel access functional description.	E
7.5.1.1	Superframe structure	N/R
7.5.1.2	Incoming and outgoing frame structure	N/R

Table 11-16 – Selections from clause 7.5 of [IEEE 802.15.4]

Clause	Title and remarks/modifications	Statement
7.5.1.3	Inter-frame (IFS) spacing – See Annex C for the inter-frame spacing description.	E
7.5.1.4	CSMA-CA algorithm – See Annex C for a description of the CSMA-CA algorithm (including priority, ARQ, segmentation and reassembly overview).	E
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 is agnostic to a physical channel.	S
7.5.2.1.1	ED channel scan – ED channel scan is not supported by the present Recommendation	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 Recommendation	N/R
7.5.2.1.4	Orphan channel scan – Orphan channel scan is not supported by the present Recommendation	N/R
7.5.2.2	PAN identifier conflict resolution PAN conflict handling is as described in clause 11.5.2.	N/R
7.5.2.2.1	Detection – PAN conflict detection is performed by scanning all incoming PAN Id of frames received by the devices as described in clause 11.5.2.	N/R
7.5.2.2.2	Resolution – On detection of a PAN identifier conflict, a device shall generate a CONFLICT frame as described in clause 11.5.2.	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 which are out of the scope of this Recommendation, 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

Table 11-16 – Selections from clause 7.5 of [IEEE 802.15.4]

Clause	Title and remarks/modifications	Statement
7.5.2.3.4	Updating superframe configuration and channel PIB attributes – The macBeaconOrder parameter shall be set to 15 to have a beaconless PAN. – The phyCurrentPage and phyCurrentChannel parameters are not used and shall be set to 0.	S
7.5.2.4	Beacon generation – Only non-beacon-enabled PAN are used – Beacon shall be transmitted using the robust modulation	S
7.5.2.5	Device discovery – Device discovery is done using the active scanning procedure described in clause 11.4.5.2.2.2, to force a coordinator to send a beacon.	E
7.5.3	Association and disassociation	N
7.5.3.1	Association – Association is fully described in clause 11.4.5.	N/R
7.5.3.2	Disassociation – Disassociation is fully described in clause 11.4.5.	N/R
7.5.4	Synchronization	N/R
7.5.4.1	Synchronization with beacons – Beacon synchronization is not used in this Recommendation.	N/R
7.5.4.2	Synchronization without beacons	N/R
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 shall begin an association procedure.	S
7.5.5	Transaction handling – Transactions are not supported in the present Recommendation.	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 Recommendation defines an acknowledgement differently. The detailed ACK implementation is described in Annex E.	E

Table 11-16 – Selections from clause 7.5 of [IEEE 802.15.4]

Clause	Title and remarks/modifications	Statement
7.5.6.4.2	Acknowledgement – The present Recommendation defines an acknowledgement differently. The detailed ACK implementation is described in Annex E.	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 – Key table contains two 16-octets keys. They represent current and preceding GMK as described in clause 12.5.3. The KeyIndex parameter selects the actual key. – Device table is not used – Security level table is not used – Automatic request attributes are not used – Default key source is not used	S
7.5.8.1.1	Key table – Key table contains two 16-octets keys. They represent current and preceding GMK as described in clause 12.5.3. The KeyIndex parameter selects the actual key.	S
7.5.8.1.2	Device table	N/R
7.5.8.1.3	Minimum security level table	N/R
7.5.8.1.4	Frame counter	N
7.5.8.1.5	Automatic request attributes	N/R
7.5.8.1.6	Default key source	N/R
7.5.8.1.7	PAN coordinator address	N/R
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/R
7.5.8.2.3	Incoming frame security procedure – The KeyIndex parameter selects the actual key from Key table.	S
7.5.8.2.4	Incoming frame security material retrieval procedure	N/R
7.5.8.2.5	KeyDescriptor lookup table	N/R
7.5.8.2.6	Blacklist checking procedure	N/R
7.5.8.2.7	DeviceDescriptor lookup procedure	N/R
7.5.8.2.8	Incoming security level checking procedure	N/R
7.5.8.2.9	Incoming key usage policy checking procedure	N/R

11.3.5.2 Extensions to clause 7.5 of [IEEE 802.15.4]: Neighbour Table

Every device shall maintain a "neighbour table" which contains information about all the devices within the POS of a device. Similarly to IEEE 802.15.4, the POS of an ITU-T G.9903 device is the reception range of an ITU-T G.9903 packet transmission. 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 shall be accessible by the adaptation, MAC sublayers and physical layer. Each entry of this table contains the fields listed in Table 11-17:

Table 11-17 – Neighbour table for CENELEC-A

Field Name	Size/Type	Description
Short address	16 bits	The MAC short address of the node which this entry refers to.
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 shall not be used.
Modulation	2 bits	Defines the modulation type to use for communicating with the device. 0x00: Robust mode 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
TXCOEF[3:0]	4 bits	Specifies the number of gain steps requested for the tones represented by TM[0] (optional)
TXCOEF[7:4]	4 bits	Specifies the number of gain steps requested for the tones represented by TM[1] (optional)
TXCOEF[11:8]	4 bits	Specifies the number of gain steps requested for the tones represented by TM[2] (optional)
TXCOEF[15:12]	4 bits	Specifies the number of gain steps requested for the tones represented by TM[3] (optional)
TXCOEF[19:16]	4 bits	Specifies the number of gain steps requested for the tones represented by TM[4] (optional)
TXCOEF[23:20]	4 bits	Specifies the number of gain steps requested for the tones represented by TM[5] (optional)
Reserved	8 bits	Reserved by ITU-T
LQI	8 bits	Link quality indicator
Age	8 bits	The remaining lifetime of the device in minutes. – When the entry is created, this value shall 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 receipt of a tone map response, this value is set to macMaxAgeTime (see Table 11-14).

Table 11-17 – Neighbour table for CENELEC-A

Field Name	Size/Type	Description
IsNeighbour	8 bits	The remaining lifetime of the validity of this entry in the table in minutes. Every time an entry is created or a frame (data or ACK) is received from this neighbour, it is set to macMaxNeighborValidTime. When it reaches zero, this entry is no longer valid in the table and may be removed.
Reserved	8 bits	Reserved by ITU-T

Table 11-18 – Neighbour Table for FCC

Field Name	Size/Type	Description
Short address	16 bits	The MAC short address of the node which this entry refers to.
ToneMap	24 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 the frequency sub-band shall not be used.
Modulation	3 bits	Defines the modulation type to use for communicating with the device. 0x00: Robust mode 0x01: DBPSK 0x02: DQPSK 0x03: D8PSK 0x04: 16-QAM (Note) 0x05-0x07: reserved by ITU-T
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
TXCOEF[1:0]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[0] (optional)
TXCOEF[3:2]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[1] (optional)
TXCOEF[5:4]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[2] (optional)
TXCOEF[7:6]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[3] (optional)
TXCOEF[9:8]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[4] (optional)
TXCOEF[11:10]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[5] (optional)
TXCOEF[13:12]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[6] (optional)

Table 11-18 – Neighbour Table for FCC

Field Name	Size/Type	Description
TXCOEF[15:14]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[7] (optional)
TXCOEF[17:16]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[8] (optional)
TXCOEF[19:18]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[9] (optional)
TXCOEF[21:20]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[10] (optional)
TXCOEF[23:22]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[11] (optional)
TXCOEF[25:24]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[12] (optional)
TXCOEF[27:26]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[13] (optional)
TXCOEF[29:28]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[14] (optional)
TXCOEF[31:30]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[15] (optional)
TXCOEF[33:32]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[16] (optional)
TXCOEF[35:34]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[17] (optional)
TXCOEF[37:36	2 bits	Specifies the number of gain steps requested for the tones represented by TM[18] (optional)
TXCOEF[39:38]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[19] (optional)
TXCOEF[41:40]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[20] (optional)
TXCOEF[43:42]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[21] (optional)
TXCOEF[45:44]	2 bits	Specifies number of gain steps requested for the tones represented by TM[22] (optional)
TXCOEF[47:46]	2 bits	Specifies the number of gain steps requested for the tones represented by TM[23] (optional)
LQI	8 bits	Link quality indicator
Age	8 bits	The remaining lifetime of the device in minutes. – When the entry is created, this value shall 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 receipt of a tone map response, this value is set to macMaxAgeTime (see Table 11-14).

Table 11-18 – Neighbour Table for FCC

Field Name	Size/Type	Description
IsNeighbour	8 bits	The remaining lifetime of the validity of this entry in the table in minutes. Every time an entry is created or a frame (data or ACK) is received from this neighbour, it is set to macMaxNeighborValidTime. When it reaches zero, this entry is no longer valid in the table and may be removed.
Reserved	8 bits	Reserved by ITU-T
NOTE – The coherent mode specified in clause 10.1.2 is optional.		

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

- Modulation = 0 (Robust mode)
- ToneMap = (all bits set to 1) AND (*adpToneMask*)
- TxGain = 0b0000
- TxCoeff = 0x0
- LQI = 0
- Age = 0
- IsNeighbour = macMaxNeighborValidTime

The neighbour table is available in the information base under the attribute *macNeighbourTable* (see clause 11.3.4.2.2).

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

The security suite specifications described in clause 7.6 of [IEEE 802.15.4] apply, with the selections specified in Table 11-19.

Table 11-19 – Selections from clause 7.6 of [IEEE 802.15.4]

Clause	Title and remarks/modifications	Statement
7.6	Security suite specification	N
7.6.1	PIB security material – Key table contains two 16 octets keys. They represent current and preceding GMK as described in clause 12.5.3. The KeyIndex parameter selects the actual key. – Automatic request attributes are not used – Default key source is not used – Device table is not used – Security level table is not used	S
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 Recommendation: 0x00 = "none", 0x05 = "ENC-MIC-32".	S

Table 11-19 – Selections from clause 7.6 of [IEEE 802.15.4]

Clause	Title and remarks/modifications	Statement
7.6.2.2.2	Key identifier mode subfield – One key identifier mode is allowed by the present Recommendation: 0x01 = "Key determined from the 1-octet Key Index subfield" The number of keys is limited to 2 (KeyIndex value is 0x0-0x1)	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 – Key index value is 0x0-0x1	N
7.6.3	Security operations	N
7.6.3.1	Integer and octet representation	N
7.6.3.2	CCM* Nonce Nonce is formatted as follows, with the first field defining the most significant byte and the last the least significant byte: • PAN-ID (2 bytes) • Source Short Address (2 bytes) • PAN-ID (2 bytes) • Source Short Address (2 bytes) • Frame Counter (4 bytes) • Security Level (1 bytes) NOTE 1 – The encrypted frame shall contain the source short address and PAN-ID in the MAC header. NOTE 2 – Fields bigger than a single byte are used in the order from the byte containing the highest numbered bits to the byte containing the lowest numbered bits (Big Endian).	S, E
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 Recommendation: 0x00 = "none", 0x05 = "ENC-MIC-32".	S
7.6.3.4.3	c data output – Two values are allowed by the present Recommendation: 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

11.3.7 Message sequence chart illustrating MAC – PHY interaction (based on clause 7.7 of IEEE 802.15.4)

11.3.7.1 Selections from clause 7.7 of IEEE 802.15.4: Message sequence chart illustrating MAC

The message sequence chart illustrating MAC – PHY interaction described in clause 7.7 of [IEEE 802.15.4] applies, with the selections specified in Table 11-20.

Table 11-20 – Selections from clause 7.7 of [IEEE 802.15.4]

Clause	Title and remarks/modifications	Statement
7.7	Message sequence chart illustrating MAC-PHY interaction – Figure 78: replaced by clause 11.3.7.2.1 – Figure 79: N/R – Figure 80: N/R – Figure 81: N/R – Figure 82: N/R – Figure 83: replaced by clause 11.3.7.2.2 – Figures 84 and 85: replaced by clause 11.3.7.2.3 – Figure 86: N/R – Additional figure about channel estimation in clause 11.3.7.2.4	S, E

11.3.7.2 Extensions to clause 7.7 of [IEEE 802.15.4]: Message sequence chart illustrating MAC

11.3.7.2.1 PAN start message sequence chart for PAN coordinators

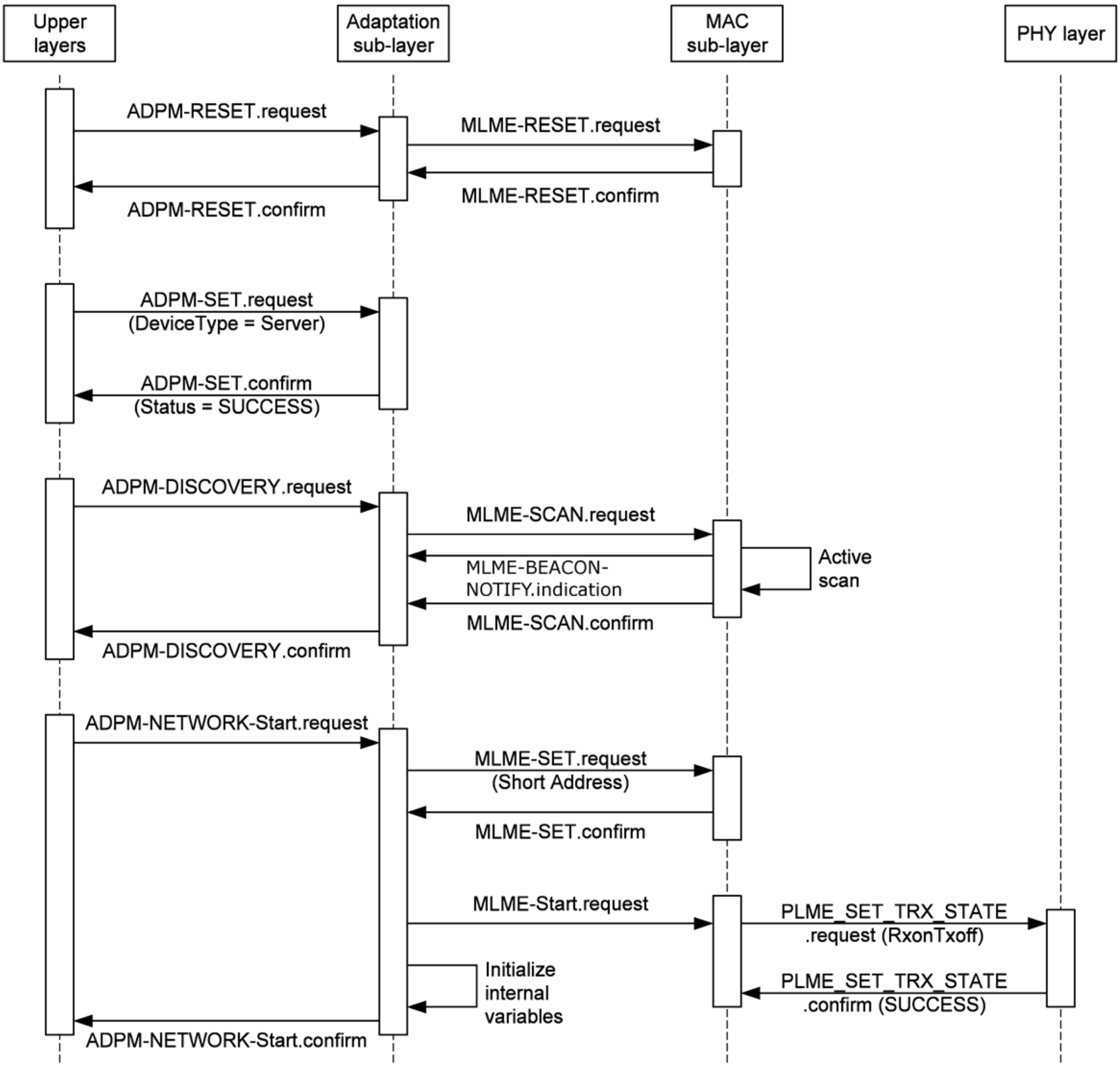


Figure 11-1 – PAN start message sequence chart

11.3.7.2.2 Active scan message sequence chart

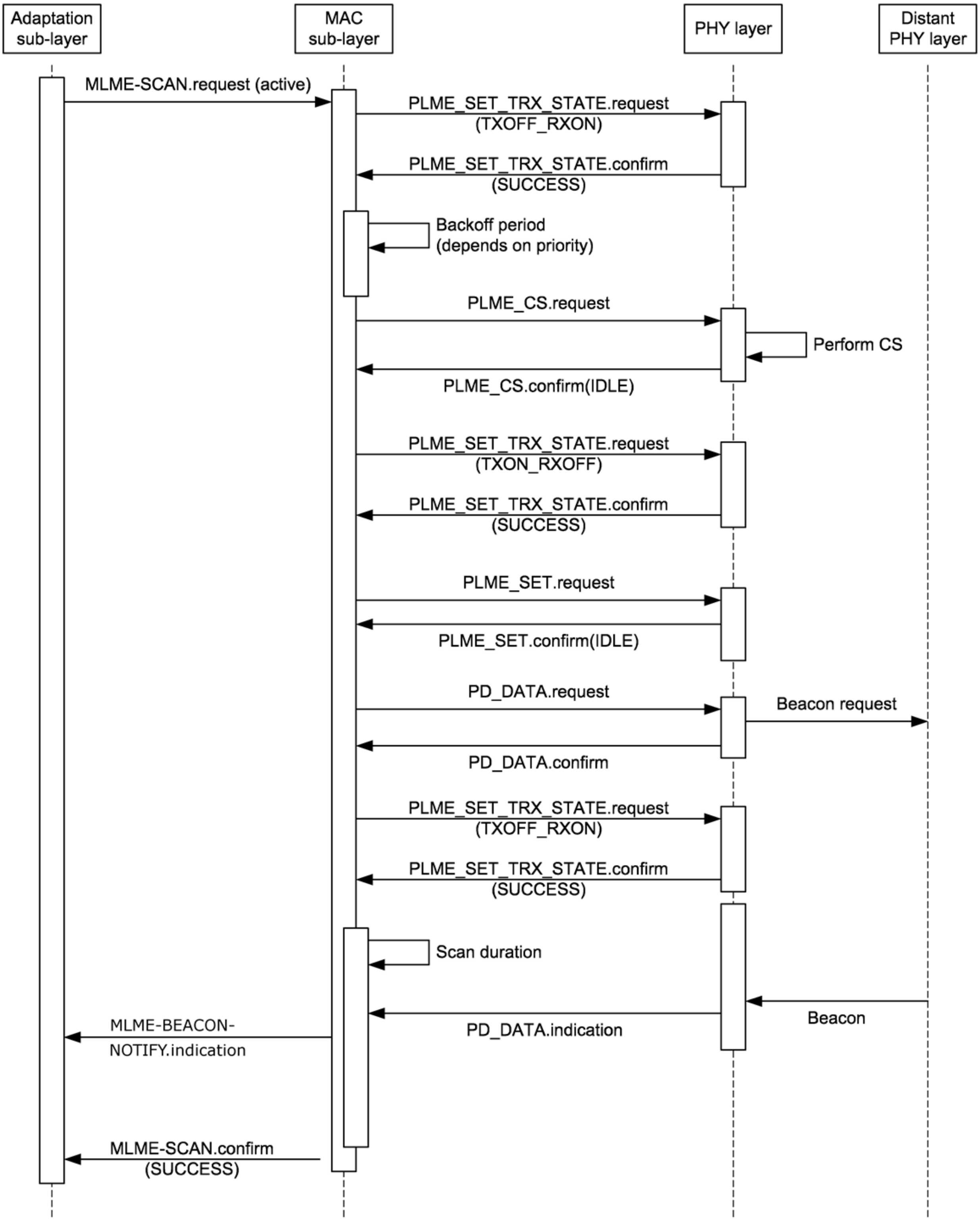

Adaptation sub-layer
MAC sub-layer
PHY layer
Distant PHY layer
MLME-SCAN.request (active)
PLME_SET_TRX_STATE.request (TXOFF_RXON)
PLME_SET_TRX_STATE.confirm (SUCCESS)
Backoff period (depends on priority)
PLME_CS.request
Perform CS
PLME_CS.confirm(IDLE)
PLME_SET_TRX_STATE.request (TXON_RXOFF)
PLME_SET_TRX_STATE.confirm (SUCCESS)
PLME_SET.request
PLME_SET.confirm(IDLE)
PD_DATA.request
Beacon request
PD_DATA.confirm
PLME_SET_TRX_STATE.request (TXOFF_RXON)
PLME_SET_TRX_STATE.confirm (SUCCESS)
Scan duration
Beacon
PD_DATA.indication
MLME-BEACON-NOTIFY.indication
MLME-SCAN.confirm (SUCCESS)

Figure 11-2 – Active scan message sequence chart

11.3.7.2.3 Data transmission message sequence chart

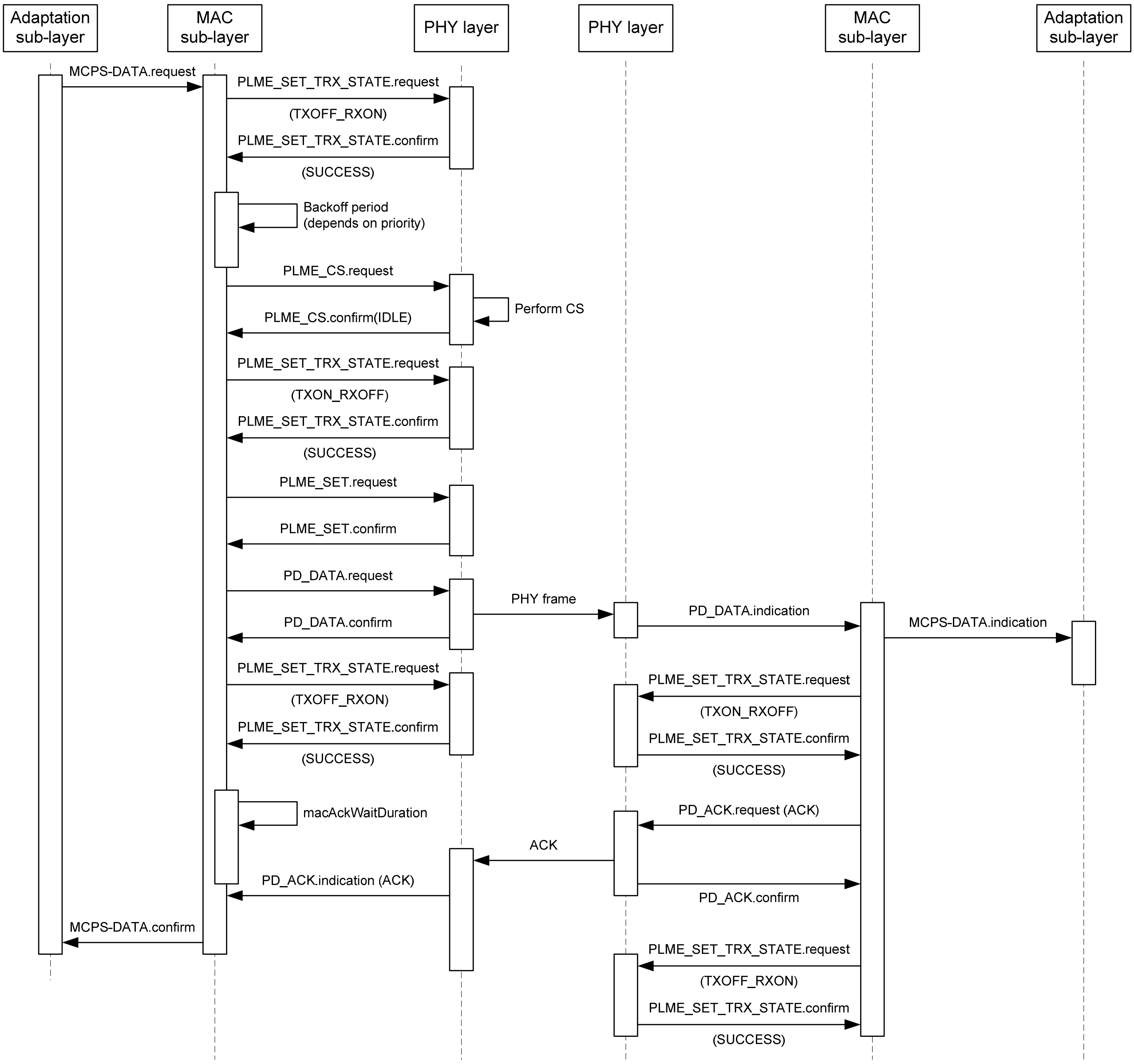


Figure 11-3 – Data transmission message sequence chart

11.3.7.2.4 Channel estimation message sequence chart

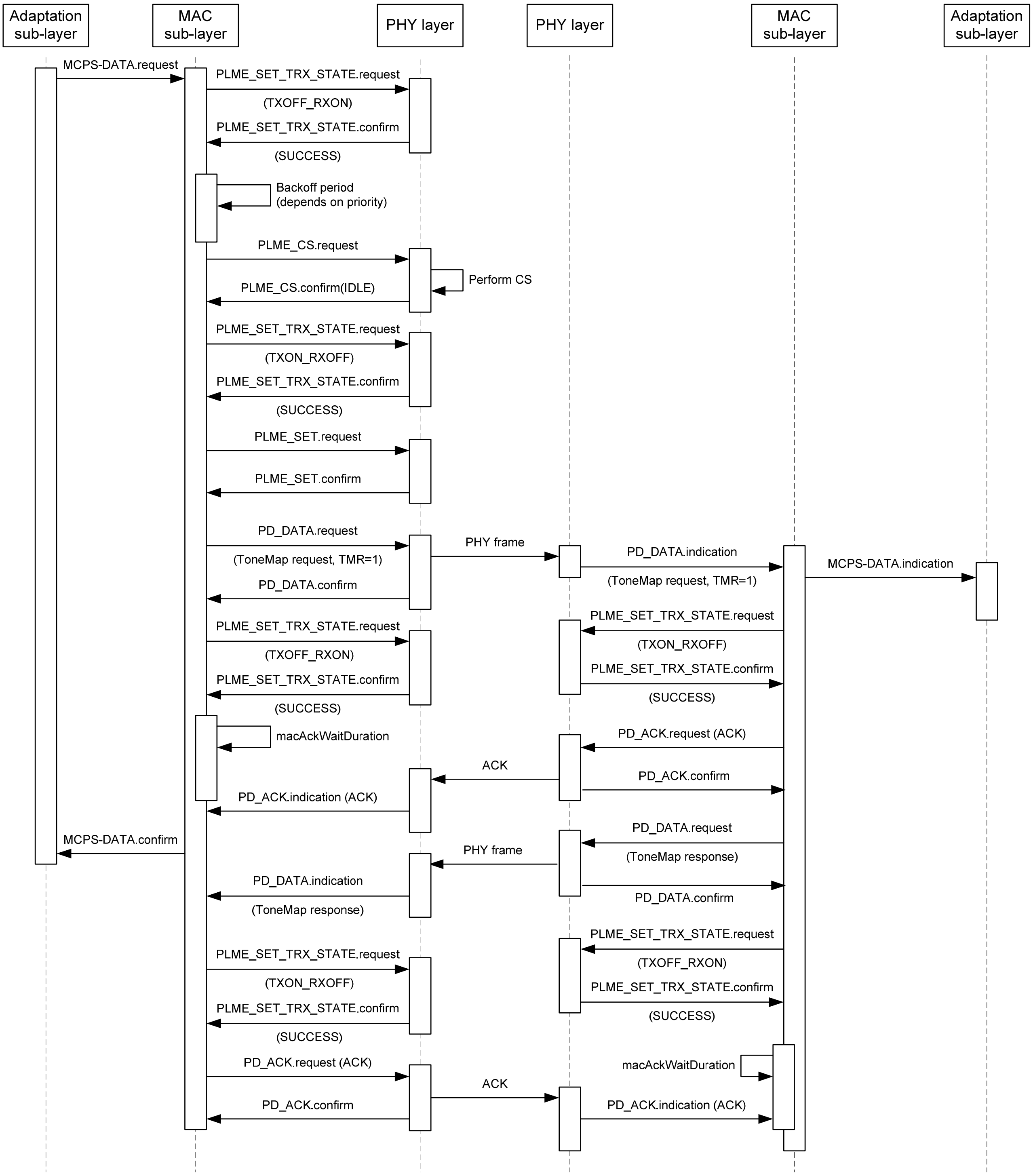


Figure 11-4 – Channel estimation (tone map request) message sequence chart

11.3.8 MAC annexes (based on IEEE 802.15.4 annexes)

The MAC annexes of [IEEE 802.15.4] apply, with the selections specified in Table 11-21.

Table 11-21 – Selections from the MAC annexes of [IEEE 802.15.4]

Clause	Title and 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 A.	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

11.4 Adaptation sublayer specification

11.4.1 Services and primitives

The services and primitives of the adaptation sublayer are described in Annex F.

11.4.2 Information base attributes

11.4.2.1 General

Table 11-22 lists the information base (IB) attributes of the adaptation sublayer.

Table 11-22 – Adaptation sublayer IB attributes

Attribute	Identifier	Type	Read only	Range	Description	Default
GMK	0x00	16 bytes	No		Write-only 16 byte GMK.	All Zero
adpIPv6Address	0x01	IPv6 address	Yes	Any	Defines the IPv6 address obtained from adpShortAddress.	FE80:::: FFFF:00FF: FE00:FFFF
adpBroadcastLogTable EntryTTL	0x02	Unsigned integer	No	0-3 600	Defines the time while an entry in the adpBroadcastLogTable remains active in the table (in seconds).	90
adpMaxBroadcastWait	0x03	Unsigned integer	No	0-3 600	Maximum wait time in seconds for broadcast packets.	90
adpMaxDiscoveryPerHour	0x04	Unsigned integer	No	0-200	Maximum number of discovery requests per hour.	60

Table 11-22 – Adaptation sublayer IB attributes

Attribute	Identifier	Type	Read only	Range	Description	Default
adpNumDiscoveryAttempts	0x05	Unsigned integer	No	0-15	Number of discovery attempts.	6
adpDiscoveryAttempts Speed	0x06	Unsigned integer	No	1-3 600	Allows programming the maximum wait time between invocation of two consecutive network discovery primitive (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 shall last (in seconds).	5
adpBroadcastLogTable	0x0B	Set	Yes	–	Contains the broadcast log table, see clause 11.4.2.2 and clause 11.4.4.2.2.1.	Empty
adpRoutingTable	0x0C	Set	Yes	–	Contains the routing table, see Table 11-17 (CENELEC-A) and Table 11-18 (FCC).	Empty
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	0-0x0E	Defines the maximum number of hops to be used by the routing algorithm.	8
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	Any	The Max duration between RREQ and the correspondent RREP (in seconds).	20
adpRrtTtl	0x13	Unsigned integer	No	0-3 600	The time to live of a route request table entry (in seconds).	90
adpKr	0x14	Unsigned integer	No	0-31	A weight factor for ROBO to calculate link cost(1).	0

Table 11-22 – Adaptation sublayer IB attributes

Attribute	Identifier	Type	Read only	Range	Description	Default
adpKm	0x15	Unsigned integer	No	0-31	A weight factor for modulation to calculate link cost[(1)].	0
adpKc	0x16	Unsigned integer	No	0-31	A weight factor for number of active tones to calculate link cost[(1)].	0
adpKq	0x17	Unsigned integer	No	0-31	A weight factor for LQI to calculate route cost[(1)].	10 for CENELEC-A, 40 for FCC
adpKh	0x18	Unsigned integer	No	0-31	A weight factor for hop to calculate link cost[(1)].	4 for CENELEC-A, 2 for FCC
adpRREQRetries	0x19	Unsigned integer	No	Any	The number of RREQ retransmission in case of RREP reception time out.	0
adpRREQRERRWait	0x1A	Unsigned integer	No	Any	The number of seconds to wait between two consecutive RREQ\RRER generations.	30
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).	3 for CENELEC-A, 5 for FCC
adpKrt	0x1C	Unsigned Integer	No	0-31	A weight factor for the number of active routes in the routing table to calculate link cost[(1)].	0
adpSoftVersion	0x1D	Set	Yes	–	The software version.	–
adpSnifferMode	0x1E	Unsigned Integer	No	0-1	Sniffer mode activation/deactivation.	0
adpMaxJoinWaitTime	0x21	Unsigned Integer	No	0-1023	Network joint timeout in seconds for LBD.	20
adpPathDiscoveryTime	0x22	Unsigned integer	No	Any	Timeout for path discovery in msec.	5 000
adpUseNewGMKTime	0x23	Unsigned Integer	No	All	The wait time in seconds for a device to use new GMK after rekeying as described in clause 12.5.4.	3 600
adpExpPrecGMKTime	0x24	Unsigned integer	No	All	The time in seconds to keep PrecGMK after switching to a new GMK as described in clause 12.5.4.	3 600

[(1)] Link cost calculation is provided in Annex B.

11.4.2.2 Routing table and broadcast table entry

Table 11-23 describes the routing table entry.

Table 11-23 – Routing table entry

Size → 16 bits	3 bits	18 bits
Next Hop address	Status	Life time (in seconds)

Table 11-24 – Broadcast log table entry

Field Name	Size	Description
SrcAddr	2 bytes	The 16-bit source address of a broadcast packet. This is the address of the broadcast initiator.
SeqNumber	Integer, 1 byte	The sequence number contained in the BC0 header.
TimeToLive	13 bits	The remaining time to live of this entry in the broadcast log table, in seconds.

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

11.4.3.1 Selections from IETF 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 are specified in [IETF RFC 4944] using the selections listed in Table 11-25.

Table 11-25 – Selections from [IETF RFC 4944]

Clause	Title and 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 shall be acknowledged – Only non-beacon-enabled networks 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 clause 11.4.3.2.1. – When more than one LoWPAN header is used in the same packet, they shall appear in the following order: Mesh addressing header Broadcast header Fragmentation header Command frame header (see clause 11.4.3.2.1)	E
5.1	Dispatch type and header	N

Table 11-25 – Selections from [IETF RFC 4944]

Clause	Title and remarks/modifications	Statement
5.2	Mesh addressing type and header – The value of the HopsLeft field shall not exceed adpMaxHops (see clause 11.4.2.1).	S
5.3	Fragmentation type and header	N
6	Stateless address auto-configuration – The interface identifier (see [IETF RFC 4291]) for an IEEE 802.15.4 interface shall be based on the EUI-64 identifier assigned to the device, the latest being itself based on an EUI-48. – Additional care shall be taken when choosing a PAN identifier, so as not to interfere with I/G and U/L bits of the interface identifier. If the PAN identifiers are chosen randomly, then they shall 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	S
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

11.4.3.2 Extensions to IETF RFC 4944

11.4.3.2.1 Command frame header

In addition of the LoWPAN header specified in [IETF RFC 4944], the present Recommendation defines a new one: command frame header. This is used for the mesh routing procedure defines in clause 11.4.4.

As shown in Figure 11-5, the ADP sublayer command frames are identified using the ESC header type (see clause 5.1 of [IETF RFC 4944]), followed by an 8-bit dispatch field indicating the type of ADP command. This header shall be in the last position if more than one header is present in the 6LowPAN frame.

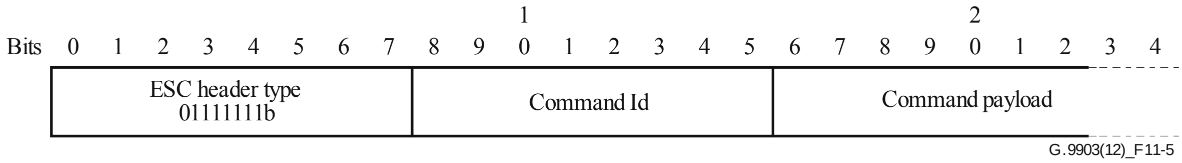


Figure 11-5 – Command frame header format

The ADP sublayer command frames are specified in Table 11-26.

Table 11-26 – Command frame header identifier

Command	Command Id	Comments	Specified in…
Mesh routing message	0x01	Use for mesh routing protocol	Clause 11.4.4
LoWPAN bootstrapping protocol message	0x02	Use for LoWPAN Bootstrap procedure	Clause 11.4.5
Contention free access command	0x03	Optional	Clause 11.4.3.2.2

11.4.3.2.2 Contention free access command

The contention free access procedure is an optional feature of this Recommendation and described in clause C.4.

The adaptation layer generates the contention free access (CFA) command if it receives an ADPD-DATA.request primitive with QualityOfService = 2 (See clause F.1.2).

Figure 11-6 defines the format of the CFA command (see also Table 11-27).

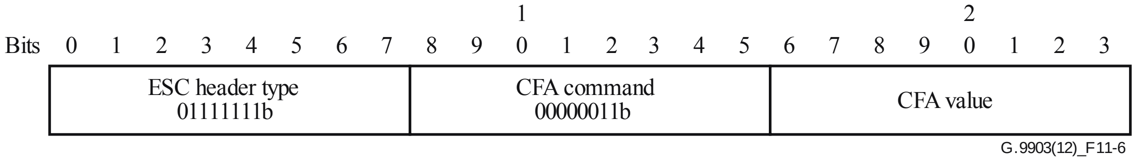


Figure 11-6 – CFA command format

Table 11-27 – 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 at the same time. Other devices shall ask the network coordinator for permission to use a contention free slot (CFS) for transmission by sending a CFA command with the request. The network coordinator may allow a requested device to use a CFS for transmission by sending a confirmation response. After receiving a successful response from the network coordinator the requested device can start a transmission during CFS. If the network coordinator denies a request the device shall not use a CFS for transmission. The requested device shall 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 the QOS parameter of the MCPS-DATA.request primitive.

11.4.4 Mesh routing (based on Annex H)

11.4.4.1 Selections from Annex H

The mesh routing as described in Annex H applies, with the selections specified in Table 11-28.

Table 11-28 – Selections from Annex H

Clause	Title and 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 B as a metric of routing.	S, E
4	Terminology	N
5	Data structures	N
5.1	Routing table entry – The destination address shall be a 16-bit address – The next hop address shall 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 shall be a 16-bit address – The reverse route address shall be a 16-bit address.	S
5.3	Message format – For the path discovery procedure, two messages have been added: path request (PREQ) and path reply (PREP). See clause 11.4.4.2.4.	E
5.3.1	Route request (RREQ) – The CT field shall be equal to 0x0F, to specify the use of the route cost described in Annex B – The D bit shall be set to 1 – The O bit shall be set to 1 – The link layer destination and originator address shall be 16-bit addresses.	S
5.3.2	Route reply (RREP) – The CT field shall be equal to 0x0F, to specify the use of the route cost described in Annex B – The D bit shall be set to 1 – The O bit shall be set to 1 – The link layer destination and originator address shall be 16-bit addresses.	S

Table 11-28 – Selections from Annex H

Clause	Title and remarks/modifications	Statement
5.3.3	Route error (RERR) – The D bit shall be set to 1 – The O bit shall be set to 1 – The unreachable address shall 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 shall repair the route locally and execute the repairing procedure described in the present clause.	S

7	Configuration parameters
	– The values of the configuration parameters shall be:
	NET_TRAVERSAL_TIME = adpNetTraversalTime
	RREQ_RETRIES = adpRREQRetries
	WEAK_LQI_VALUE = adpWeakLQIValue
	– Extension: the following parameters are added by the present Recommendation:
	RREQ_RERR_WAIT = adpRREQRERRWait
	PATH_DISCOVERY_TIME = adpPathDiscoveryTime
		S, E

8	IANA consideration	N
9	Security considerations	N/R
10	Acknowledgements	N/R
11	References	N
11.1	Normative reference	N
11.2	Informative reference	I

11.4.4.2 Extensions to Annex H

11.4.4.2.1 Unicast packet routing

The routing of the unicast packet is performed using the following algorithm on receipt of an 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 clause F.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 clause D.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 H.6.5
  - 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 11.4.4.2.2

ELSE

- Drop the frame

11.4.4.2.2 Multicast/broadcast

11.4.4.2.2.1 Packet routing

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

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

Each node shall have a broadcast log table. This table is used for routing broadcast packets and each entry contains the parameters described in Table 11-24.

Each time a device receives a broadcast address with a HopsLft field of a mesh header (see clause 5.2 of [IETF RFC 4944]) strictly greater than 0, it shall 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. Otherwise, a new entry is added in the table and the TimeToLive field is initialized with the value adpBroadcastLogTableEntryTTL (see clause 11.4.2). 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 shall decrement its HopsLft field. If HopsLft is not zero, it triggers the transmission of the received broadcast frame.

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

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

- IF ((SrcAddr, SeqNumber) exists in broadcast log table)
  - Discard frame
- ELSE
  - Create one entry (SrcAddr, SeqNumber, adpBroadcastLogTableEntryTTL) in the broadcast log table, with the corresponding frame characteristics.
  - Generate an ADPD-DATA.indication primitive to the upper layer with the following characteristics:
    - DstAddrMode = 0x02
    - 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 clause D.2)
    - SecurityEnabled = (SecurityLevel != 0)
  - HopsLft=HopsLft –1
  - If (HopsLft > 0), Trigger the frame transmission.

NOTE – In case of a multicast address, the broadcast address 0xFFFF is used at the MAC level as mentioned in clause 3 of [IETF 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 clause 11.4.2).

11.4.4.2.2.2 Groups

Each device can belong to one or more groups of devices. The IB attribute adpGroupTable (see clause 11.4.2) 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 clause 11.4.4.2.2.1).

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

11.4.4.2.3 Route discovery

11.4.4.2.3.1 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 an RREQ frame and executes the algorithms as described in clause 11.4.4.1.

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

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

All devices shall handle RREQ, RREP and RERR frames as described in clause 11.4.4.1 and modify their routing tables accordingly.

11.4.4.2.3.2 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 clause 11.4.4.1 in order to find a route to the destination. If the route discovery succeeds, then the data frame is sent to the destination according to the newly discovered route. If the route discovery fails, then the adaptation layer shall 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 shall generate an ADPD-DATA.confirm primitive with the status code ROUTE_ERROR.

Route repairing procedures are described in clause 11.4.4.1.

11.4.4.2.3.3 RREQ RERR generation frequency limit

A node shall wait RREQ_RERR_WAIT second between two successive RREQ/RERR generations to limit the number of broadcast packets in the network. The definition of the RREQ_RERR_WAIT parameter is given in clause 11.4.4.1.

11.4.4.2.4 Path discovery

11.4.4.2.4.1 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 the following sub-clauses.

After the algorithm is completed (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 at the same time. Any 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 clause 11.4.4.2.4.2).

Once transmitted, the node waits for a path reply (PREP), otherwise, 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 towards the destination. Else, the node just discards the PREQ.

Generating a path reply (PREP)

Upon receiving a path request (PREQ), a final node 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 towards 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 is no route to the destination, the node just discards the path reply message received.

11.4.4.2.4.2 Path request frame

The path request frame format and the detail of its related fields are described in Figure 11-7 and Table 11-29 respectively.

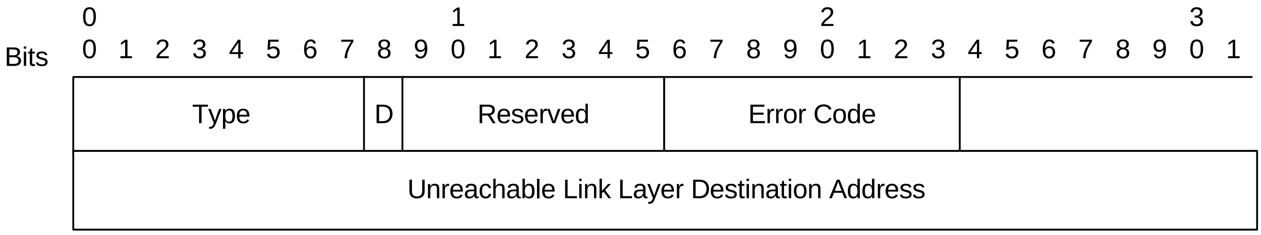


Figure 11-7 – Path request (PREQ) message format

Table 11-29 – Path request (PREQ) field definitions

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 the node which originated the packet.

11.4.4.2.4.3 Path reply frame

The path reply frame format and the detail of its related fields are described in Figure 11-8 and Table 11-30 respectively.

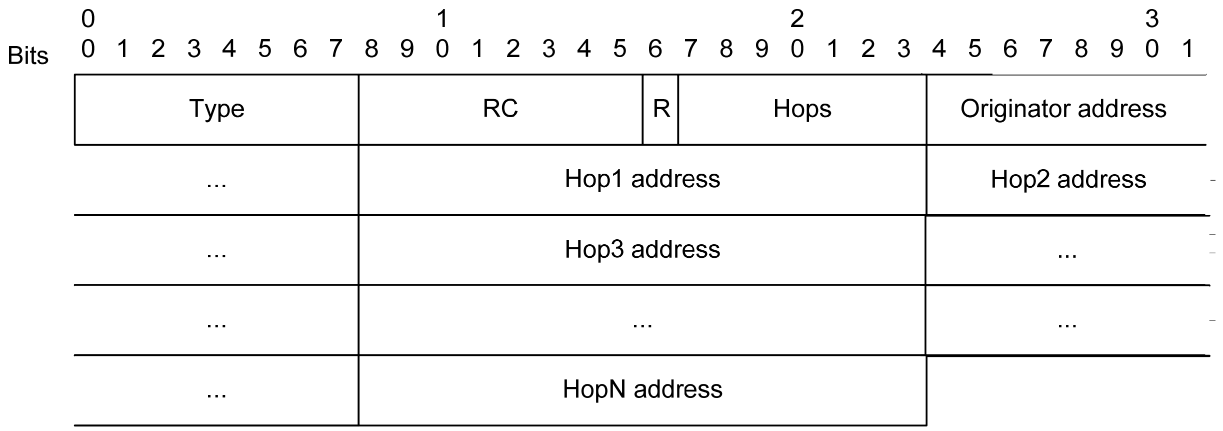


Figure 11-8 – Path reply (PREP) message format

Table 11-30 – Path reply (PREP) field definitions

Field	Size (bits)	Value	Definition
Type	8	5	Path reply (PREP) 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 the node which originated the packet.
Hop$_N$	16	–	The 16 bit short link layer address of nodes constituting the path.

11.4.5 Commissioning of new devices (based on Annex J)

11.4.5.1 Selections from Annex J

The commissioning of new devices on an existing network described in Annex J applies, with the selections specified in Table 11-31.

Table 11-31 – Selections from Annex J

Clause	Title and 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 shall perform an active scan.	S

Table 11-31 – Selections from Annex J

Clause	Title and remarks/modifications	Statement
3.3	LoWPAN bootstrapping mechanism – 'LBA discovery phase' is described in clause 11.4.5.2.2	E
3.3.1	LoWPAN bootstrapping protocol message format	N
3.3.1.1	LBP message – Some enhancements and clarifications to the LBP message format are given in clause 11.4.5.2.1.	E
3.3.2	LoWPAN bootstrapping information base – PAN_type shall be secured. – Address_of_LBS shall be equal to the default address of the PAN coordinator that is 0x0000. – Short_Addr_Distribution_Mechanism shall be 0 for centralized address management.	S
3.3.3	LBA discovering phase – Some enhancements and clarifications to 6LoWPAN bootstrapping procedure are given in clause 11.4.5.2.2. – The LBD shall perform an active scan instead of broadcasting an LBA solicitation message.	E
3.3.4	LoWPAN bootstrapping protocol (LBP)	S
3.3.5	Bootstrapping in open 6LoWPAN	N/R
3.3.6	1. The LBP messages from the LBD to the LBA are sent by invocation of the MCPS-DATA.request primitive with the following attributes: – SrcAddrMode = 0x03 – SrcAddr = Own EUI-64 address – DstAddrMode = 0x02 – DstAddr = 16-bit short address of the LBA passed as an argument to the ADPM-NETWORK-JOIN.request primitive – DstPANId = The PAN ID passed as an argument to the ADPM-NETWORK-JOIN.request primitive – msduLength = length of the LBP message – msdu = the LBP message itself – msduHandle = random number – QualityOfService = 0 – SecurityLevel = FALSE. All other parameters can be ignored. 2. The LBP messages from the LBA to the LBS are relayed by using the routing algorithm as described in clause 11.4.4 with the DiscoverRoute parameter set to TRUE and the SecurityLevel set to a non-zero value. 3. The LBP messages in LBS are sent by invocation of the ADPM-LBP.request primitive which carries the following attributes: – DstAddrType = 0x02 – DstAddr = 16 bit LBA address – NsduLength = the length of the LBP message – Nsdu = the LBP message itself – NsduHandle = random number – NsduType = 0	S

Table 11-31 – Selections from Annex J

Clause	Title and remarks/modifications	Statement
	– MaxHops = maximum number of hops – DiscoverRoute = TRUE – QualityOfService = 0 – SecurityEnabled = TRUE 4. The LBP messages from the LBS to the LBD are relayed to the LBA by using the routing algorithm as described in clause 11.4.4 with the DiscoverRoute parameter set to TRUE and the SecurityLevel set to a non-zero value. 5. The LBP messages from the LBA to the LBD are sent by invocation of the MCPS-DATA.request primitive with the following attributes: – SrcAddrMode = 0x02 – SrcAddr = Own 16-bit short address – DstAddrMode = 0x03 – DstAddr = The EUI-64 contained as a LBD in the LBP message – DstPANId = LBA PAN_ID – msduLength = length of the LBP message – msdu = the LBP message itself – msduHandle = random number – QualityOfService = 0 – SecurityLevel = FALSE.	
3.3.7	Role of entities in LBP – If an LBD does not find any LBA during the LBA discovery phase, it shall still perform LBA discoveries as long as it is not commissioned. Note that LBA discovery is done using active scans rather than 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 an IPv6 address – The devices do not need to obtain an IPv6 address prefix and the procedures described in this clause as well as in [IETF RFC 4862] shall be ignored. Only the IPv6 link local address generated as stated in clause 7 of [IETF RFC 4944] is used for communication.	M

3.6	Configuration parameters
	– The values of the configuration parameters shall be:
	CHANNEL_LIST = 0xFFFF800 (not used)
	SCAN_DURATION = adpActiveScanDuration (see clause 11.4.2.1)
	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

Table 11-31 – Selections from Annex J

Clause	Title and remarks/modifications	Statement
7	Acknowledgements	N/R
8	References	N
8.1	Normative references	N
8.2	Informative references	I

11.4.5.2 Extensions to Annex J

11.4.5.2.1 LoWPAN bootstrapping protocol (LBP) message format

11.4.5.2.1.1 General

The LBP message format and the details of its related fields and its parameters are described in Figure 11-9, Table 11-32 and clause 11.4.5.2.1.3 respectively.

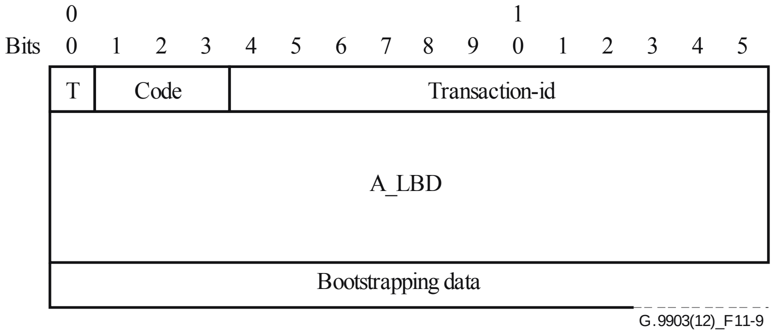


Figure 11-9 – LBP message format

Where

T identifies the type of message (1-bit)

0 Message from LBD

1 Message to LBD

Code identifies the message code (3-bit) defined in Table 11-32.

Transaction-id aids in matching responses with requests (12-bit)

A_LBD The A_LBD 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 clause 11.4.5.2.1.2)

configuration parameters (see clause 11.4.5.2.1.3).

Table 11-32 – T and code fields in LBP message

T	Code	LBD message	Description
0	001	JOINING	The LBD requests joining a PAN and provides the necessary authentication material.
1	001	ACCEPTED	Authentication succeeded with delivery of device specific information (DSI) to the LBD
1	010	CHALLENGE	Authentication in progress. PAN specific information (PSI) may be delivered to the LBD
1	011	DECLINE	Authentication failed
0/1	100	KICK	KICK frame is used by a PAN coordinator to force a device to lose its MAC address, or by any device to inform the coordinator that it left the PAN. On receipt of this frame, a device shall 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 clause 11.4.5.2.2.7 for 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 clause 11.5.2 for details about PAN ID conflict handling.

11.4.5.2.1.2 Embedded EAP messages

LBP messages embed Extended Authentication messages (EAP) as defined in [IETF RFC 3748]. Figure 11-10 describes minor modification to fit the generic LBP information element format.

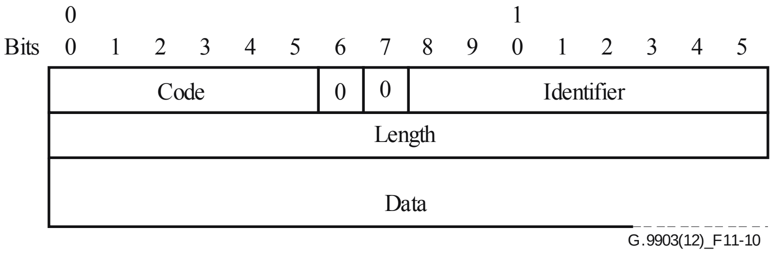


Figure 11-10 – Embedded EAP message format (generic)

where

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

1 Request (sent to the peer = LBD)

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 [IETF RFC 3748]. The conversion appears straightforward in both directions. The proper conversion shall apply when the EAP message is propagated over another protocol (i.e., RADIUS) and in case of integrity protection covering the EAP header.

Identifier 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 shall 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 [IETF RFC 3748] for more details on:

a specific format for request/response messages and the introduction of the type field (Identity, "Nak", etc.);

a specific format for success/failure messages with an empty data field.

11.4.5.2.1.3 Configuration parameters

The configuration parameter format and the detail of its related fields are described in Figure 11-11.

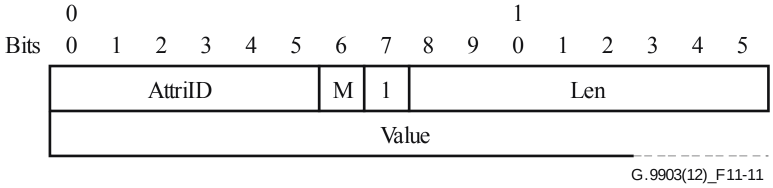


Figure 11-11 – Configuration parameter format

Where

Attr-ID represents the ID of the attribute in the LoWPAN information base (LIB) (6-bit)

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

- device specific information (DSI)
- 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.

11.4.5.2.2 6LoWPAN bootstrapping procedures

11.4.5.2.2.1 Overview

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

Figure 11-12 provides an overview of the messages exchanged between devices during the bootstrapping procedure.

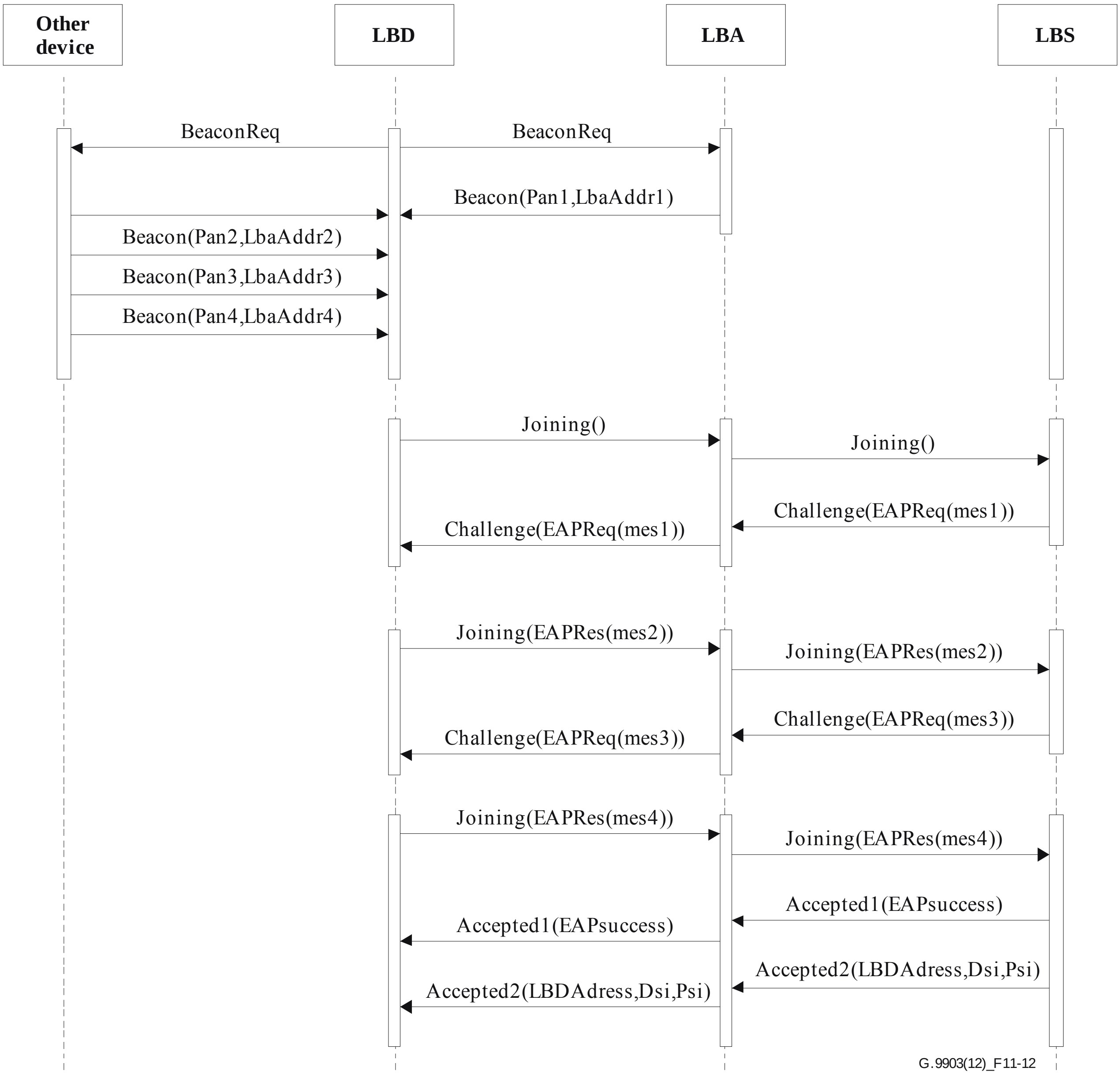


Figure 11-12 – Bootstrapping protocol messages sequence chart

Figure 11-13 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.

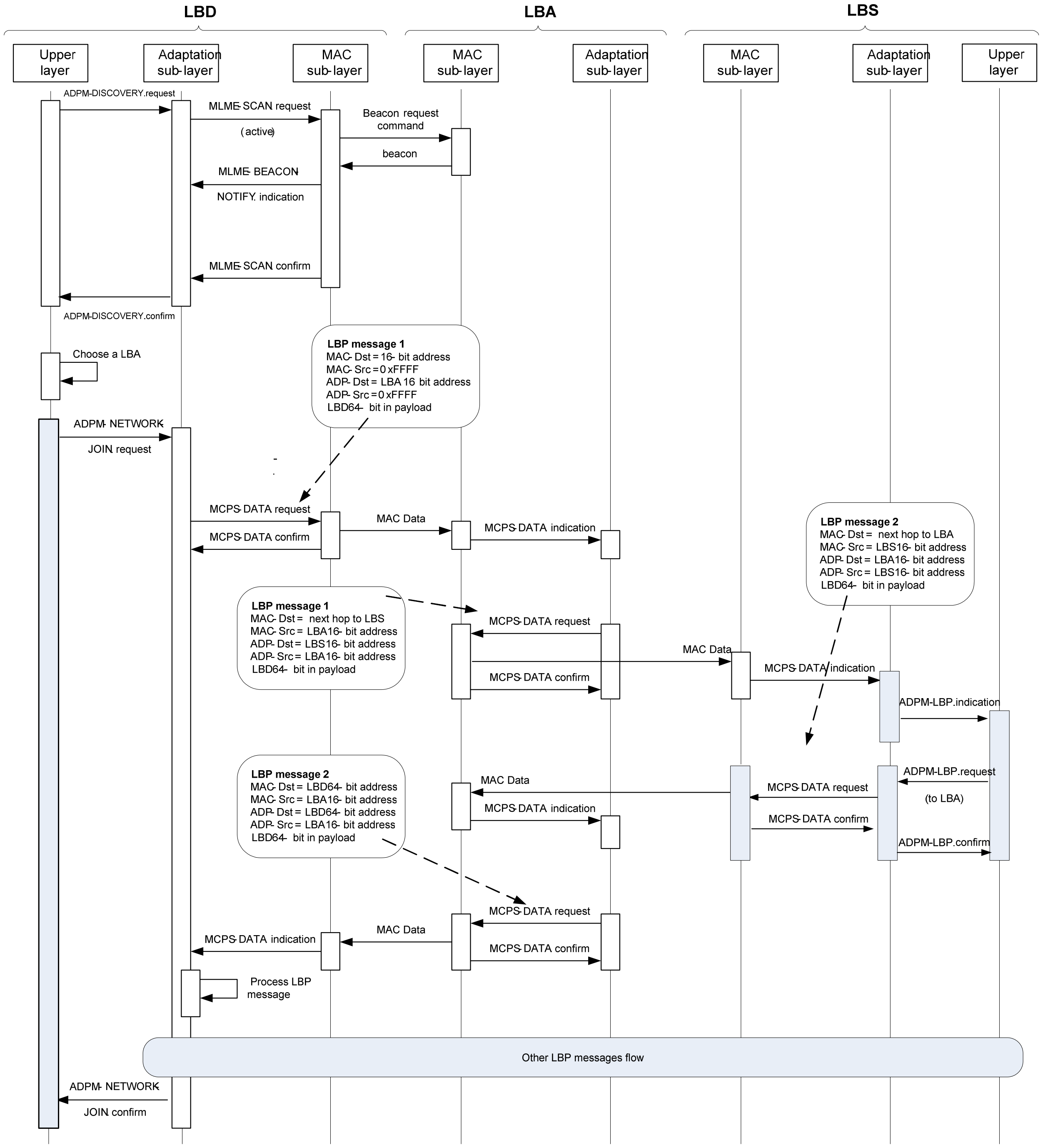


Figure 11-13 – Bootstrapping protocol messages forwarding

11.4.5.2.2.2 Discovering phase

At the beginning of the bootstrapping procedure, an end device (also known as LoWPAN bootstrapping device or LBD) shall launch an "active channel scan" (see [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 by 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 shall reply by sending a beacon frame with its PAN identifier, short address and capabilities.

Upon receiving each beacon frame, the MAC layer in the LBD issues an MLME-BEACON-NOTIFY.indication primitive with the PANDescriptor parameters corresponding to the beacon. At the end of scan duration, the adaptation layer generates an ADPM-DISCOVERY.confirm primitive which contains the PANDescriptorList. If multiple beacons with the same PAN identifier are received, only one beacon per discovered PAN is selected to be included in PANDescriptor.

The choice of the beacon is based on the following criteria:

- association permit, rejected if negative
- minimum value of route cost to coordinator
- maximum value of beacon link quality
- short address, according to a round robin algorithm.
- After finishing the scan procedure, the device can join the network following the procedure described in clause 11.4.5.2.2.6.

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

11.4.5.2.2.3 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. It is assumed that the LBA is 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.) shall be loaded in the LBS; or

– the authentication 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, [IETF 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 may 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 sends 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 Recommendation, the EAP identity phase is skipped as proposed by [IETF 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 LBD.

11.4.5.2.2.4 Authentication and key distribution phase

The authentication phase is wholly dependent on the EAP method in place. The EAP protocol is very flexible and supports 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;
- the second one for mutual control of session keys derivation.

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

- Transient EAP key (TEK) for the end-to-end security of EAP messages. These TEKs are generated as described in [IETF RFC 4764].
- Group session keys for a basic PAN security. These keys are shared by all the authenticated nodes in the PAN. Every MAC data frame with the SecurityEnabled field set to 1, except those involved in the initial phases of the bootstrapping procedure, is securely transmitted with encryption and decryption at every hop. These group keys may be refreshed periodically or when a node is detached from the PAN.

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

Refer to clause 12.5 for further details on the proposed EAP method.

11.4.5.2.2.5 Authorization and initial configuration phase

Upon completion of the authentication and key distribution process, the LBS shall send back an LBP DECLINE message embedding an EAP failure message if the authentication has failed. This message is relayed by the LBA to the LBD to inform the LBD that the LBS did not accept the join request of LBD.

If the LBS accepts the join request of the LBD, it selects a 16-bit short address, globally defined and fully routable in the PAN and sends back an LBP ACCEPTED message, embedding an EAP success message. Upon receipt of this message, the LBD activates the GMK key. A second LBP ACCEPTED message is sent by the LBS embedding the global 16-bit short address and optionally other device specific and PAN specific parameters. These messages are relayed by the LBA to the LBD. At this stage, the LBD owns a 16-bit short address and a session key allowing the secure transmission of the messages within the PAN.

Upon receipt of the LBP message, the LBD may set up an optimized route to the LBS with the help of the LOAD protocol (see clause 11.4.4). The path used by the device during association phase may be stored in routing tables as an initial route between LBD and LBS without invoking the LOAD protocol.

11.4.5.2.2.6 Joining a PAN for any node except coordinator

The network joining procedure is performed by a device which is not a PAN coordinator and does not have a short address (default short address of a device upon reset is 0xffff which means no short address). During this procedure, the device is authenticated, associated with the network and receives GMK and a short address assigned by the coordinator. After a successful discovery phase as described in clause 11.4.5.2.2.2, this procedure may be triggered by invocation of the ADPM-NETWORK-JOIN.request primitive with PANID and LBAAddress of one of the discovered devices as listed in PANDescriptor of ADPM-DISCOVERY.confirm. The selection criteria of PAN and LBAAddress is implementation specific.

If the join is successful, the upper layer is informed by a successful ADPM-NETWORK-JOIN.confirm which includes the assigned short address and PAN ID of the network. In case of failure, this procedure may be repeated after repeating the discovery phase.

If the join procedure is not complete within *adpMaxJoinWaitTime*, a fail confirmation shall be sent to the upper layer.

The upper layer in the LBS receives the join request of a device as well as subsequent authentication messages as ADPM-LBP.indication primitives with embedded LBP messages. It also sends the authentication messages as well as acceptance and short address embedded in LBP messages to the LBD by invoking ADPM-LBP.request. The processing of received LBP messages and the construction of LBP messages to be sent to the LBD is performed in the upper layer of the LBS. This includes the processing and construction of EAP messages as described in clause 12.5.

In the LBD, all the LBP messages are processed internally and the upper layer is not aware of the message exchanges during the authentication procedure. Upon completion or timeout, the upper layer of the LBD receives an ADPM-NETWORK-JOIN.confirm with success or failure status.

11.4.5.2.2.7 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 T field set to 1 and its code field set to 100b. The bootstrapping data in that message shall be empty.

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

- acknowledge the frame if necessary;
- set its 16-bit short address to 0xFFFF;
- generate an ADPM-NETWORK-LEAVE.indication containing the 64-bit address of the device;
- invoke an MLME-RESET.request primitive with the SetDefaultPIB parameter set to TRUE;
- invoke its ADPM-RESET.request primitive to reset itself.

Figure 11-14 describes the messages exchanged during the removal of a device from the PAN by the coordinator.

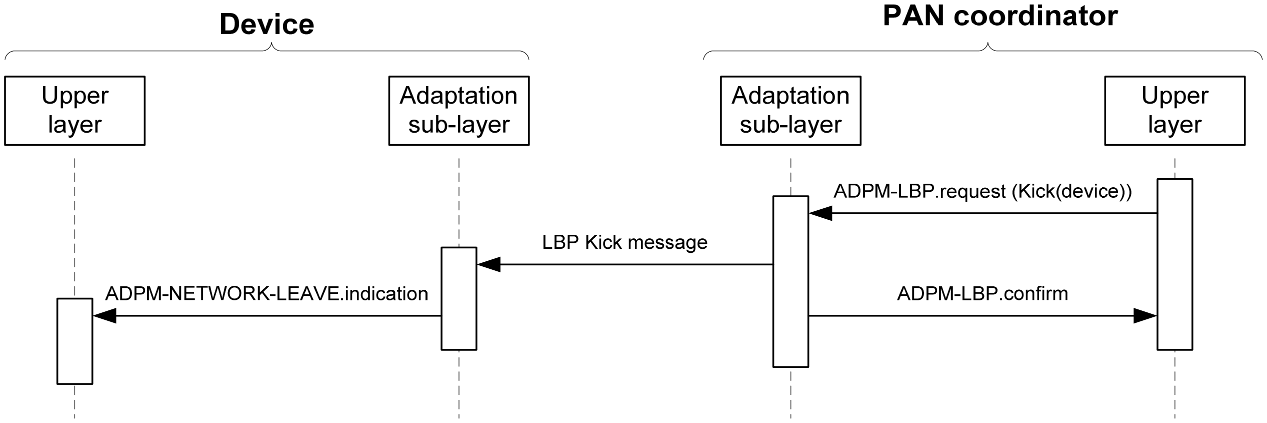


Figure 11-14 – Message sequence chart during the removal of a device by the coordinator

Upon completion of this procedure, the device shall restart the joining network procedure described in clause 11.4.5.2.2.

11.4.5.2.2.8 Leaving a PAN – Removal of a device by itself

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

If the ADPM-NETWORK-LEAVE.request primitive is invoked by a device which is the PAN coordinator, or if the ExtendedAddress parameter is not NULL, then the adaptation sublayer shall issue an ADPM-NETWORK-LEAVE.confirm primitive with the status INVALID_REQUEST.

If the ADPM-NETWORK-LEAVE.request primitive is invoked by a device which is not the PAN coordinator and the ExtendedAddress parameter is NULL, then the adaptation sublayer shall:

– Send a KICK frame to the PAN coordinator using an ADPD-DATA.request primitive using a standard LBP message with KICK frame as described in Table 11-32 with T field set to 0 and its Code field set to 100b. The bootstrapping data in that message should be empty.

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

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

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

Figure 11-15 describes the messages exchanged during the removal of a device initiated by the device itself.

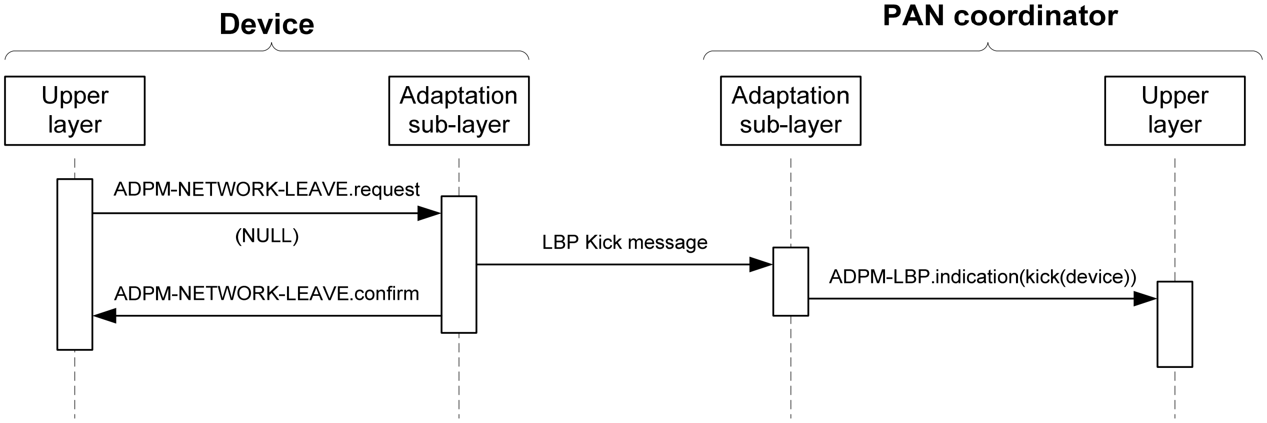


Figure 11-15 – Message sequence chart during the removal of a device by itself

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

11.4.6 Sniffer mode

This mode is used to support monitoring of the transmitted packets on the power line. Once activated, the modem will process all packets regardless of their destination address. The sniffer modem shall generate an ADPD-DATA.indication for any received packet. The modem activated in sniffer mode shall not forward packets. If a sniffer modem receives a fragment, it shall add an IPv6 fragment header to the packet so the upper layer can detect it. The fragment offset field and the identification field shall be set to the offset of the LOWPAN header and the Datagram_Tag respectively.

11.5 Functional description

11.5.1 Network formation

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

Prior to the network formation, the PAN coordinator shall perform an active scan as described in clause 11.4.5.2.2.2. 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 may 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 shall 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 this PAN ID is chosen and set in the coordinator is implementation specific.

NOTE – The PAN identifier shall be logically ANDed with 0xFCFF, as described in clause 6 of [IETF RFC 4944]; see also Table 11-25.

Once the PAN identifier has been determined, the adaptation sublayer shall 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, shall be set to 0;
- BeaconSecurityLevel = 0;
- BeaconKeyIdMode, BeaconKeySource, BeaconKeyIndex: not used, shall be set to 0.

The MAC sublayer then generates an 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.

11.5.2 PAN ID conflict detection and handling

When a device is associated with a PAN, its MAC sublayer shall analyse the destination and source PAN identifier in the MAC header of any frame it receives, at any time.

If a frame containing a destination or source PAN identifier is received and does not match the PAN identifier of the device or the 0xFFFF PAN ID, the frame should be dropped and it shall generate an 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 an MLME-SYNC-LOSS.indication primitive with another LossReason than PAN_ID_CONFLICT, it shall ignore it.

In response, the adaptation layer shall 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 shall contain the PAN Id of the detected PAN using the format defined in clause 3.3.1 of Annex J and described in Figure 11-16:

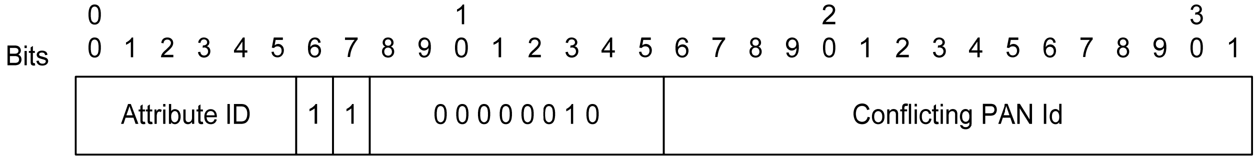


Figure 11-16 – CONFLICT message format

This frame is sent to the PAN coordinator with short address of 0x0000 using an ADPD-DATA.request primitive which carries the following attributes:

– NsduLength = the length of the frame

– Nsdu = the frame

– NsduHandle = a random number

– DiscoverRoute = TRUE

– QualityOfService = 0

– SecurityEnabled = TRUE.

This shall be translated to MCPS-DATA.request with:

– DstAddrMode = 0x02

– DstAddr = the short address of coordinator 0x0000.

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

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

- the status field set to PAN_ID_CONFLICT; and
- the AdditionalInformation field set to the conflicting PAN Id.

12 Security

12.1 Access control and authentication

An end device (ED) may not access the network without a preliminary identification (with comparison to white or black lists) and authentication. Identification and authentication are based on two parameters that personalize every ED:

- an EUI-48 MAC address as defined in [IEEE 802-2001]. This address may be easily converted into an EUI-64 as requested by [IEEE 802.15.4] and related documents.
- A 128-bit shared secret (also known as pre-shared key or PSK) used as a credential during the authentication process. It is shared by the ED itself (also known as peer) and an authentication server. The mutual authentication is based on proof that the other party knows the PSK. It is of the highest importance that 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 clause 11.4.5) that embeds the extensible authentication protocol (EAP) (see clause 11.4.5.2.1.2).

As shown in Figure 12-1, the LBP and EAP have been designed to be relayed by intermediates nodes. Then during the bootstrapping phase, if an ED (also known as LBD) that has not yet acquired a routable 16-bit address is at a 1-hop distance from the PAN coordinator (also known as LBS) they can communicate directly. Otherwise, they shall use an intermediate node (also known as LBA) located at a 1-hop distance of the LBD.

Moreover, two different authentication architectures shall be considered:

- The authentication server function is directly supported by the LBS and in this case all the authentication material (access lists, credentials, etc.) shall 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 [IETF RFC 2865]).

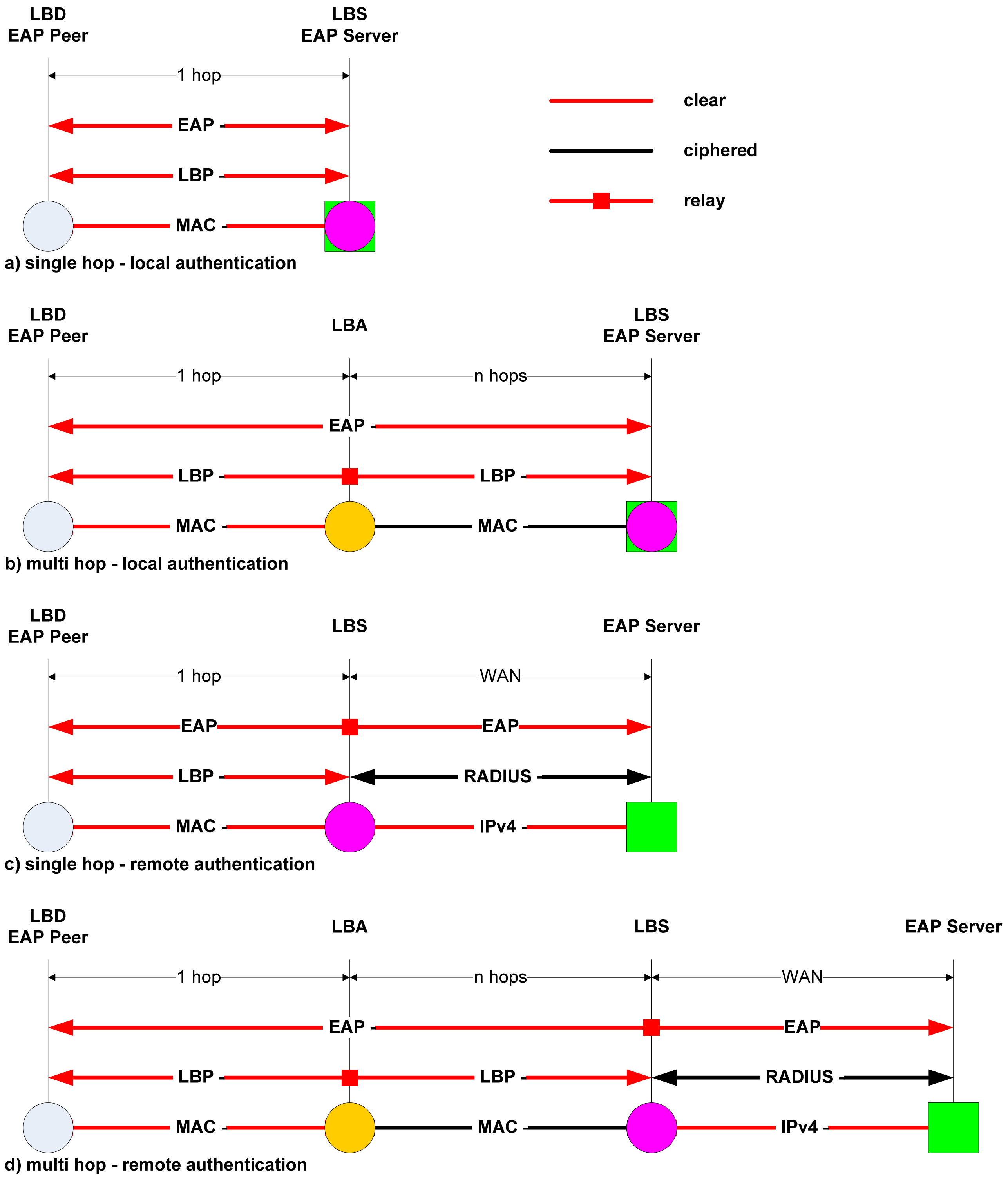


Figure 12-1 – LBP and EAP relaying capabilities

The authentication process is wholly dependent on the EAP method in place. The EAP protocol is very flexible and supports 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 clause 12.5), 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 clause 12.5.2).

12.2 Confidentiality and integrity

As shown by Figure 12-2, confidentiality and integrity services are ensured at different levels:

- At the 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 is 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 master key (GMK). This GMK is individually and securely distributed to every node by using the EAP-PSK secure channel.

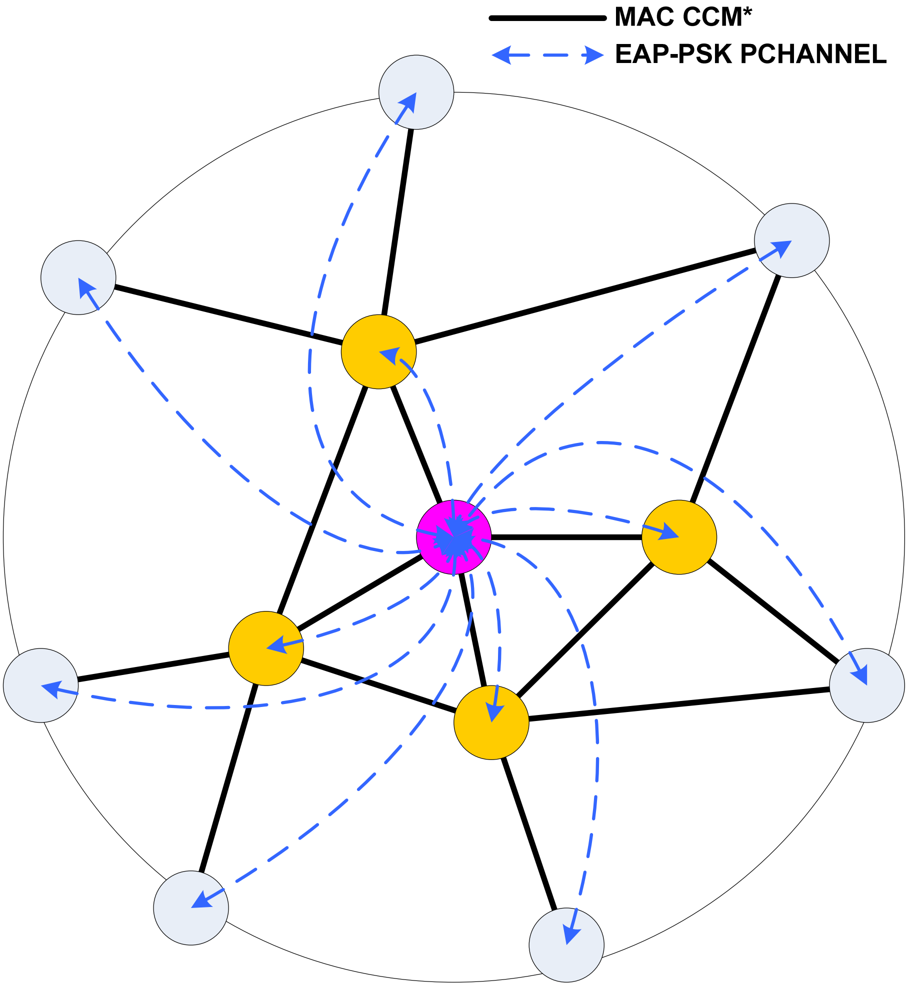


Figure 12-2 – Confidentiality and security

- At the EAP-PSK level: as defined in [IETF RFC 4764], the EAP-PSK provides confidentiality and integrity (and replay protection) services, also known as protected Channel (PCHANNEL) to the messages exchanged over the EAP between the EAP server and any peer.

12.3 Anti-replay and DoS prevention

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

The CCM* ciphering mode is generalized at the 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.

12.4 Authentication and key distribution protocol – Selections from IETF RFC 3748

Authentication and key distribution are supported by the extensible authentication protocol (EAP) as given in [IETF RFC 3748] together with the selections listed in Table 12-1.

Table 12-1 – Selections from [IETF RFC 3748]

Clause	Title and remarks/modifications	Statement
1	Introduction	N
2	Extensible authentication protocol (EAP) – Initial identity request (allows roaming and EAP method negotiation) is for further study and shall be by-passed.	S
2.1	Support for sequences	N
2.2	EAP multiplexing model – Only one EAP method is defined (cf. 11.4.5).	S
2.3	Pass-through behaviour – Over the LBP, the code field is slightly different from a regular EAP code field as specified in [IETF RFC 3748]. The conversion appears straightforward in both directions. The proper conversion shall 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 IEEE 802	N/R
3.4	Lower layer indications	N
4	EAP packet format – Over the LBP, the code field is slightly different from a regular EAP Code field.	S

Table 12-1 – Selections from [IETF RFC 3748]

Clause	Title and remarks/modifications	Statement
4.1	Request and response – Over the LBP, the code field is slightly different from a regular EAP code field.	S
4.2	Success and failure – Over the LBP, the code field is slightly different from a regular EAP code field.	S
4.3	Retransmission behaviour	N
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 clause 12.5). 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 [RFC 2284]	I

12.5 EAP method

The EAP protocol is very flexible and supports 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 the pre-shared key EAP method (EAP-PSK) as given in [IETF RFC 4764] together with the selections listed in Table 12-2.

Table 12-2 – Selections from [IETF RFC 4764]

Clause	Title and 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 with [IETF RFC 4764]. See clause 12.5.2.	N

Table 12-2 – Selections from [IETF RFC 4764]

Clause	Title and remarks/modifications	Statement
5	EAP-PSK message format – EAP-PSK extension capabilities are used for group key distribution in full compliance with [IETF RFC 4764]. See clause 12.5.2.	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

12.5.1 Overview of the EAP-PSK

According to the EAP specification the EAP-PSK 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 the EAX mode of operation as a cipher suite.

Master session key (MSK) A session key derived between the EAP peer and server. The 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. The EAP-PSK generates a 512-bit EMSK. It is not used in the OFDM CPL and shall not be generated.

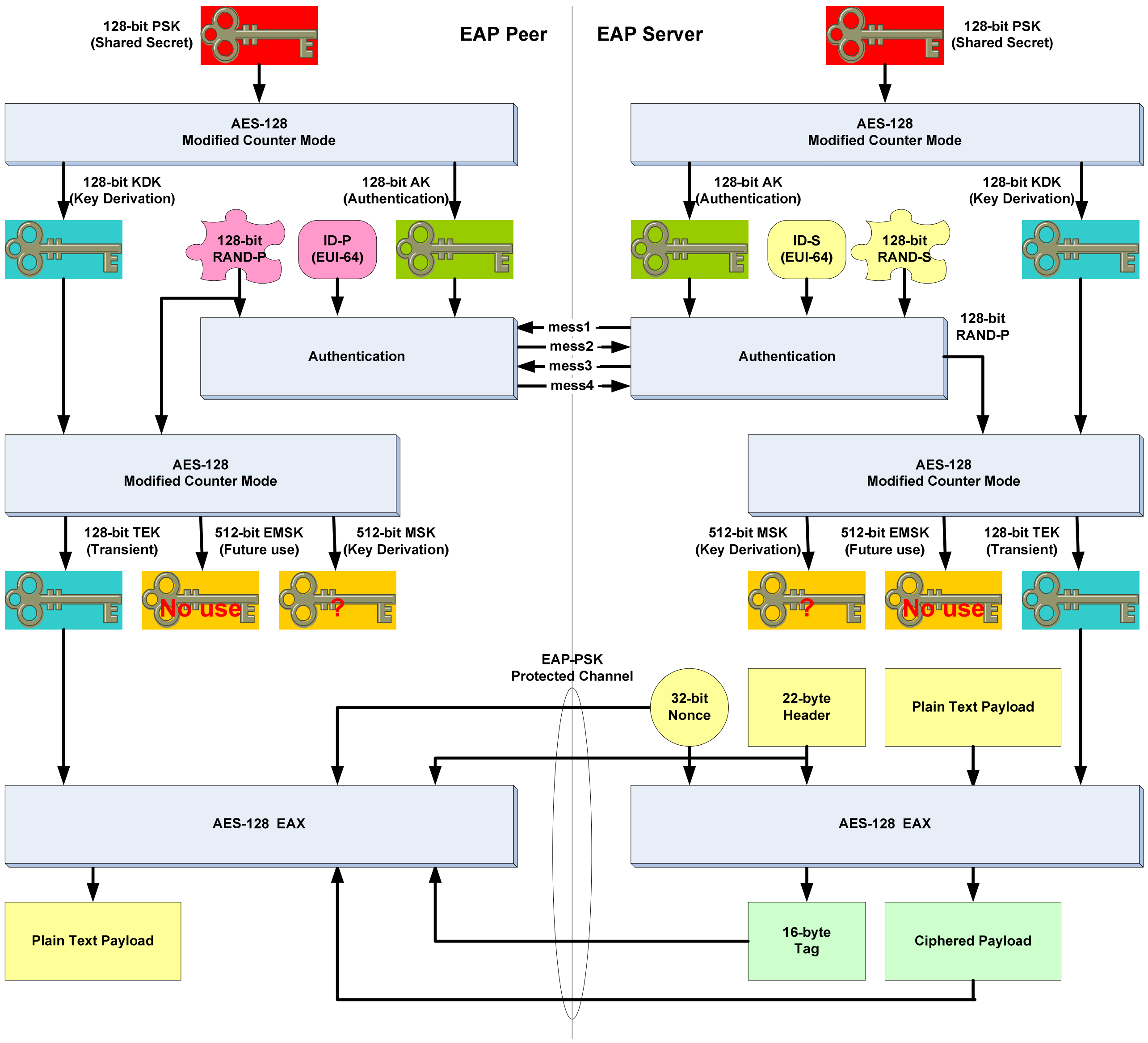


Figure 12-3 – EAP-PSK key hierarchy overview

12.5.2 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 to be random. GMK generation is considered as purely implementation dependent.

GMK is distributed to the peer in two circumstances:

- during the bootstrapping process, carried as a regular extension to EAP-PSK message 3 of Figure 12-3;
- during the re-keying process, carried as a regular extension to EAP-PSK message 5 of Figure 12-3. The GMK lifetime is rather long (several 10s years) due to the 4 byte counter included in the nonce. Nevertheless it is good policy to re-key the network regularly, or when a node is leaving it.

12.5.3 GMK field format

The GMK field in message 3 or 5 of Figure 12-3 is defined in compliance with the generic extension field (EXT) (see [IETF RFC 4764] clause 5.3.) described in Figure 12-4.

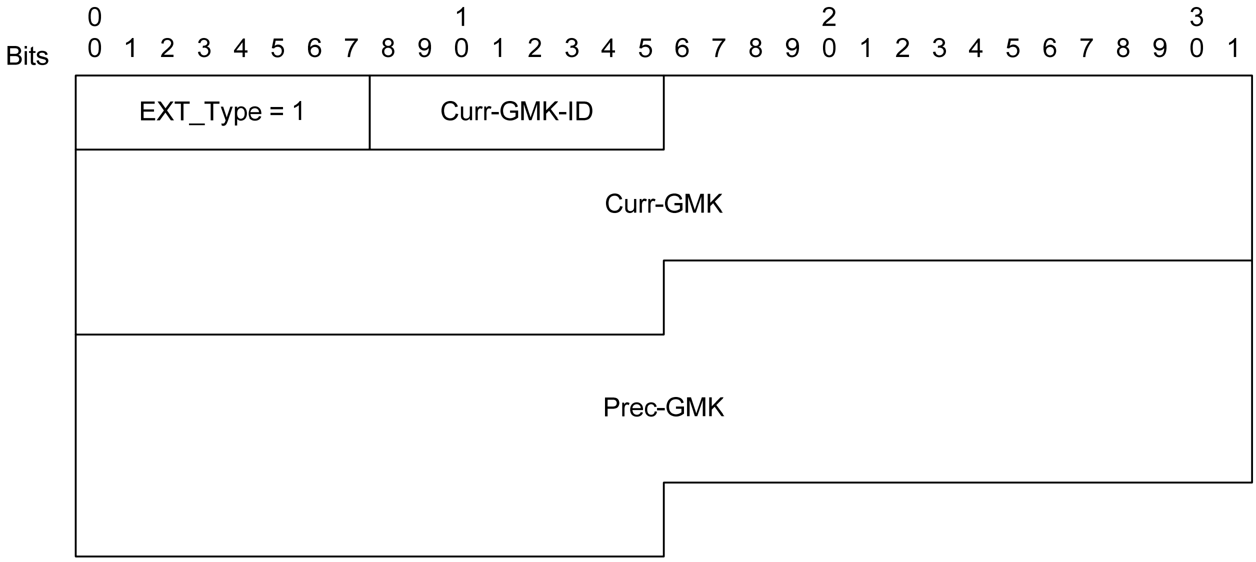


Figure 12-4 – GMK field format for messages 3 and 5

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.

12.5.4 Peer side procedure

Once a peer receives a GMK field embedded in a message 3 (in case of bootstrapping) or a message 5 (in case of re-keying), it shall install Curr-GMK in a table at Curr-GMK-ID index and Prec-GMK at its corresponding index location (the Curr-GMK-ID of the last re-keying procedure) and shall process and respond to the message according to [IETF RFC 4764].

Both keys are immediately available to decrypt a received packet using the key identified by 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 Curr-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.

In case the peer did not receive the EAP success message after a *adpUseNewGMKTime*, it may start using Curr-GMK to send frames as soon as it receives a packet with the key identifier of Curr-GMK which indicates the server had already invoked switching to the Curr-GMK in the network.

After switching to the Curr-GMK, a peer may keep receiving some messages encrypted with the Prec-GMK during a transient period. The Prec-GMK may be deleted after *adpExpPrecGMKTime* delay which shall be long enough to make sure that all devices in the network have switched to Curr-GMK.

12.5.5 Server side procedure

The bootstrapping procedure is defined in clause 11.4.5.2.2.

In case of re-keying, the EAP server generates a new GMK. Then it transmits an LBP challenge message, embedding an EAP request message that contains the newly generated GMK as Curr-GMK and the previous GMK as Prec-GMK, to every formerly associated peer. The Curr-GMK-ID shall be chosen to be different from a key identifier associated with Prec-GMK.

If the EAP server does not receive the EAP response to the LBP challenge message from a peer, it may retry the re-keying procedure described above. The PAN coordinator may remove the peer device using the procedure described in clause 11.4.5.2.2.7 if the EAP server has failed to receive the response.

Once the EAP server has received the EAP response to the LBP challenge message from all peers, it may start sending EAP success messages to the peers to invoke switching to the new GMK.

Annex A

Protocol implementation conformance statement

(This annex forms an integral part of this Recommendation.)

A.1 Overview

Compliance with clauses of [IEEE 802.15.4] shall be consistent with the extensions and selections defined in this annex.

The first part of this annex entirely takes as reference the protocol implementation conformance statement of [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 ITU-T G.9903 have been taken into account.

A.2 PICS proforma tables

A.2.1 Functional device types (from Annex D.7.1 of IEEE 802.15.4)

Table A.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		

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

Table A.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

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

Table A.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		

A.2.4 Radio frequency (from Annex D.7.2.3 of IEEE 802.15.4)

Table A.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

A.2.5 MAC sublayer functions (from Annex D.7.3.1 of IEEE 802.15.4)

Table A.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		

Table A.5 – PICS – MAC sublayer functions (from Annex D.7.3.1 of [IEEE 802.15.4])

Item number	Support			Comments
	N/A	Yes	No	
MLF9.2		X		
MLF9.2.1		X		
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	

A.2.6 MAC frames (from Annex D.7.3.2 of IEEE 802.15.4)

Table A.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 clause 7 and Annex E.
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	

Annex B

Routing Cost

(This annex forms an integral part of this Recommendation.)

This part describes the characteristics that a routing cost used in the LOAD routing algorithm (described in Annex H and in clause 11.4.4) shall have.

A route cost is defined as the sum of all the link costs on the route. As described in Annex H, a route cost is an integer value between 0 and 255, lower values meaning better routes. While the link cost computation algorithm is implementation dependent, the following formula may be used:

MOD_Kr = 1 for ROBO, 0 for other modulations

MOD_Km = 3 for ROBO, 2 for DBPSK, 1 for DQPSK and 0 for D8PSK

Link Cost = AdpKr× MOD_Kr

+ AdpKm× MOD_Km

+ AdpKc ×(Maximum Number of Tones – number of active tones)/Maximum Number of Tones

+ AdpKq × (Maximum LQI –LQI)/Maximum LQI

+ AdpKh × 1

+ AdpKrt × number of active routes/Maximum number of active routes

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 $C\{\lfloor D_i, D_j \rfloor\}$ 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 may take into account PHY transmission parameters, the number of hops, etc. The link cost computation algorithm is implementation dependent.

Annex C

Channel access

(This annex forms an integral part of this Recommendation.)

C.1 Overview

The channel access is accomplished by using the carrier sense multiple access with collision avoidance (CSMA/CA) mechanism with a random back-off time. The random back-off 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 back-off, the device shall transmit its data. If the channel is found to be busy following the random back-off, 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. The physical carrier sense (PCS) is provided by the PHY as described in clause C.3. In the latter case, the PCS shall stay high long enough to be detected and the 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 the received packet or upon collision. In these cases, the virtual carrier sense tracks the expected duration of the b'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 something other than an 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.

C.2 Inter-frame (IFS) spacing

The time intervals between frames on the medium constitute the inter-frame space and are necessary due to propagation and processing times. As shown in Figure C.1, three inter-frame space values are defined. Contention inter-frame space (CIFS) occurs after the end of the previous transmission. The second defined interval is the response inter-frame 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 inter-frame 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 inter-frame 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:

$$\text{aEIFS} = \text{aSymbolTime} \times (\text{N}_{\text{FCH}} + \text{aMaxFrameSize} + \text{aCIFS} + \text{aRIFS}) + \text{aAckTime}$$

where N_{FCH} is the number of symbols in the frame control header (FCH).

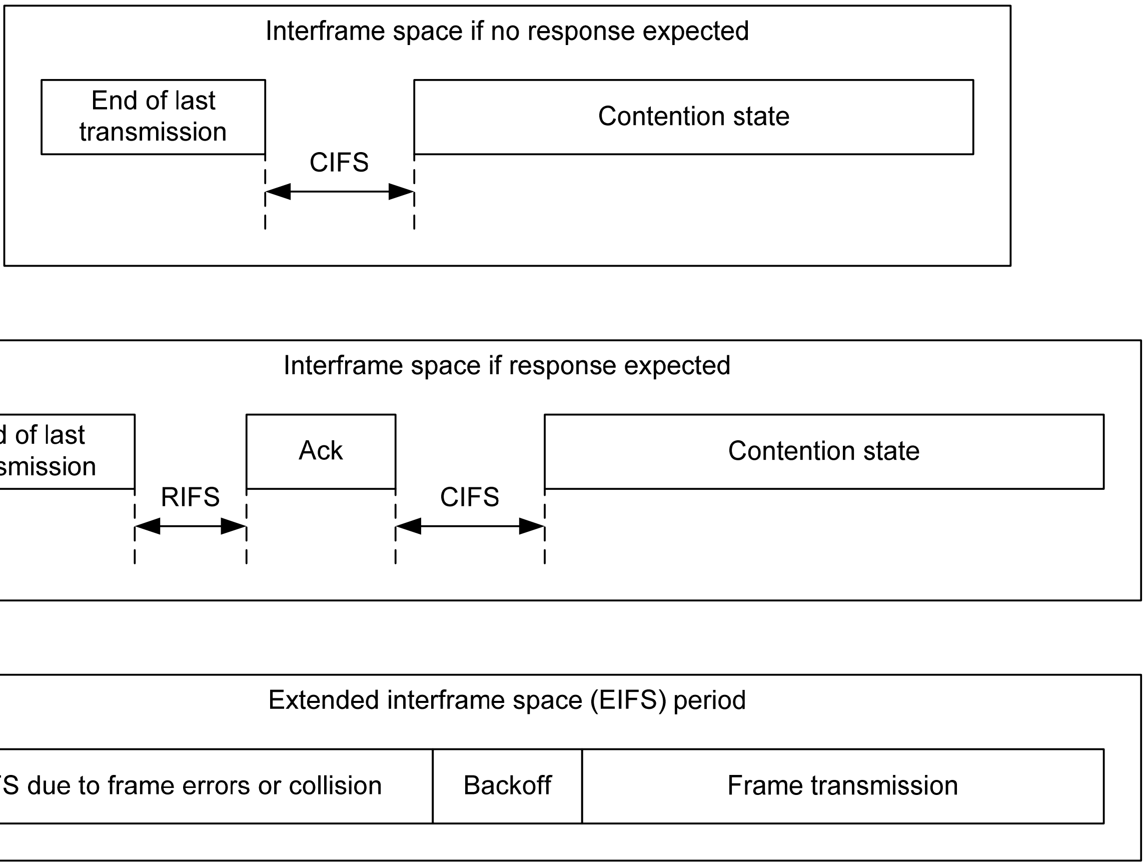

Interframe space if no response expected
End of last transmission
CIFS
Contention state
Interframe space if response expected
End of last transmission
RIFS
Ack
CIFS
Contention state
Extended interframe space (EIFS) period
EIFS due to frame errors or collision
Backoff
Frame transmission

Figure C.1 – IFS

C.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 back-off mechanism spreads the time over which stations attempt to transmit, thereby reducing the probability of collision, using a truncated binary exponential back-off 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 back-off periods, where one back-off period shall be equal to *unitBackoffPeriod* 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 has been used as back-off while attempting the current transmission; this value shall be initialized to 0 before each new transmission attempt.

BE is the back-off exponent, which is related to how many back-off 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 C.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 back-off 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)].

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

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 shall 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.

In order to improve the fairness, *BE* shall be reduced to *minBE* if *mod*(*NB, macCSMAFairnessLimit)=0* where *mod* is modulo operation. For example, if *macCSMAFairnessLimit* = 15 and *maxCSMABackoffs* = 50, as soon as the number of back-offs reaches 15, 30 and 45, the *BE* should be reduced to minBE.

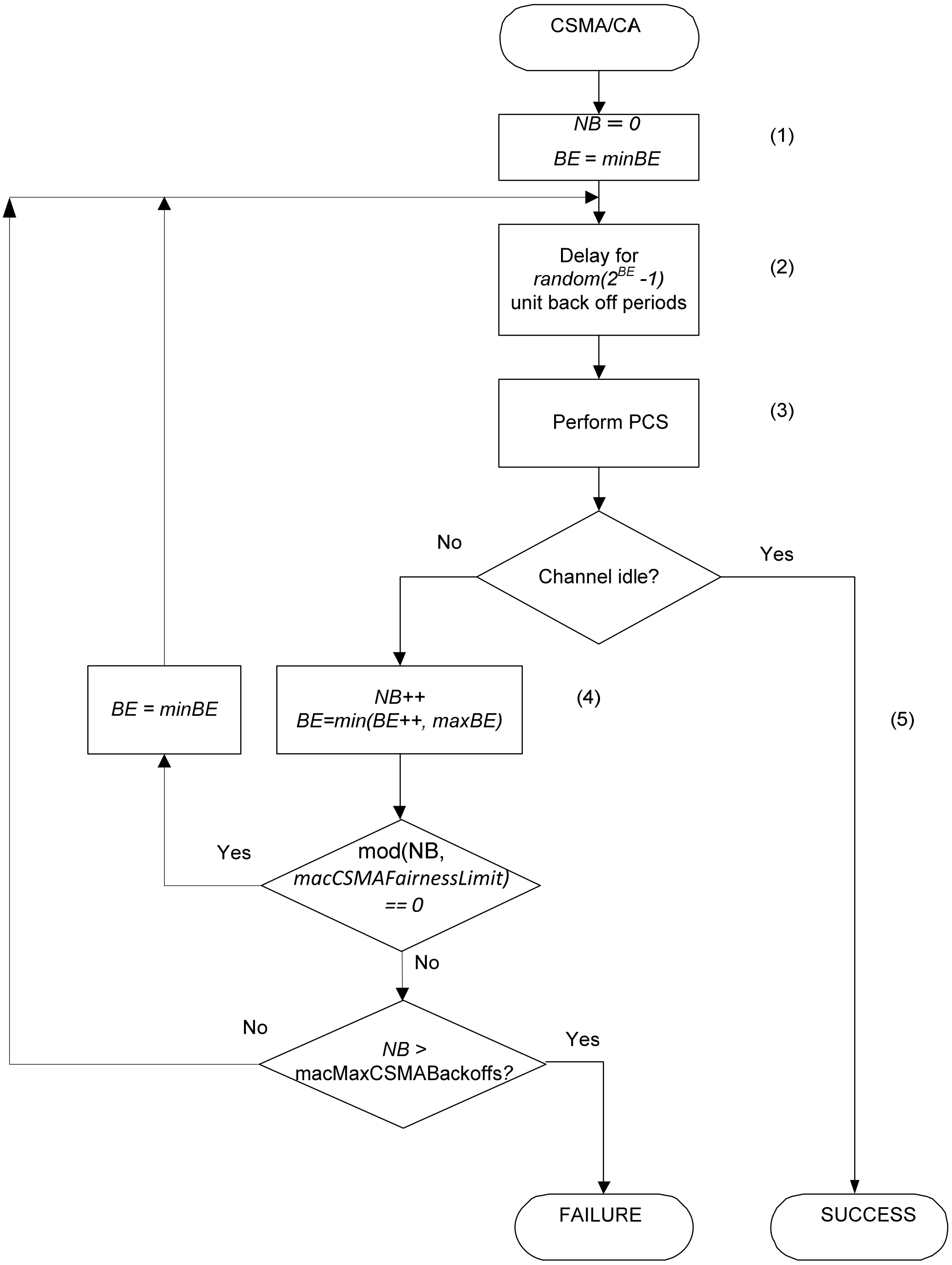


Figure C.2 – CSMA/CA algorithm

C.4 Priority

Prioritized access to the channel can be beneficial for real time application or control application when an urgent message shall 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 the contention state as shown in Figure C.3.

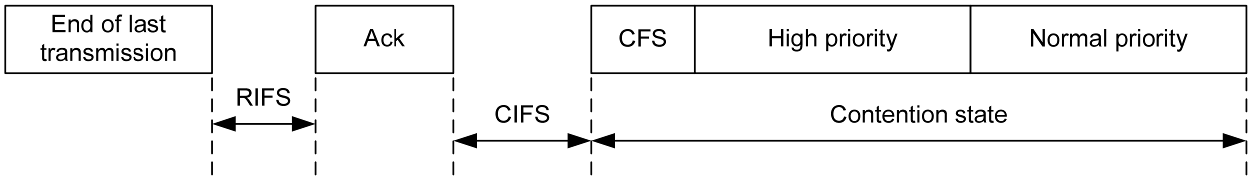


Figure C.3 – Priority contention windows

The first slot of contention window is called the contention free slot (CFS). The contention free slot shall be used for transmission of subsequent segments of a MAC packet without the back-off procedure to prevent possible interruption from other nodes and to simplify the MAC packet reassembly procedure on a receiver. In this case, only the first segment is sent using either a normal or high priority contention window and the rest are sent using the contention free slot.

The high and normal priority stations will compete for channels during the high priority contention window (HPCW) and normal priority contention window (NPCW) correspondingly. Since HPCW is located before NPCW, high priority stations will get access to the channel before the station with normal priority. The duration of HPCW and NPCW are calculated as follow:

HPCW time = macHighPriorityWindowSize × aSlotTime;

NPCW time = $(2^{maxBE} \times aSlotTime)$ – HPCW time;

CFS time = *aSlotTime*;

C.5 ARQ

The automatic repeat request (ARQ) 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 acknowledgement that when received allows the transmitter to assume successful delivery of the frame. The negative acknowledgement (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 acknowledgement. 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 acknowledgement after a waiting period, it assumes that the transmission was unsuccessful and retries the frame transmission. If an acknowledgement is still not received after several retries, the originator can choose either to terminate the transaction or to try again. When the acknowledgement is not requested, the originator assumes the transmission was successful. Also if acknowledgement is not requested the originator can retransmit the same packets few times to increase probability of data delivery. The receiver shall 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 the original.

The acknowledgement cannot be requested for broadcast or multicast transmission. On the transmit side the ARQ shall configure the number of retransmissions (cf. macMaxFrameRetries from clause 7.4.2 of [IEEE 802.15.4]) as shown in Figure C.4.

On the receive side the ARQ generates acknowledgement for the PLC packet with the correct FCS (CRC16) if the packet corresponds to this address as shown in Figure C.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 the FCS from ACK for identification.

MAC acknowledgement is described in details in Annex E.

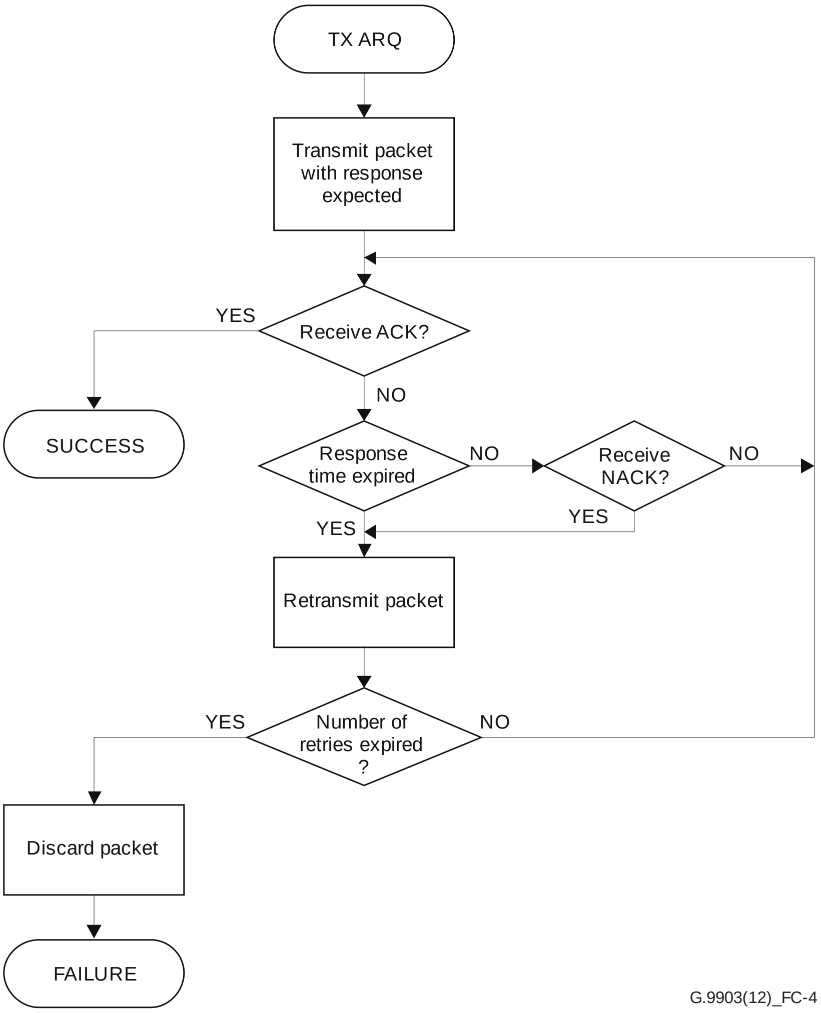


Figure C.4 – Transmit ARQ

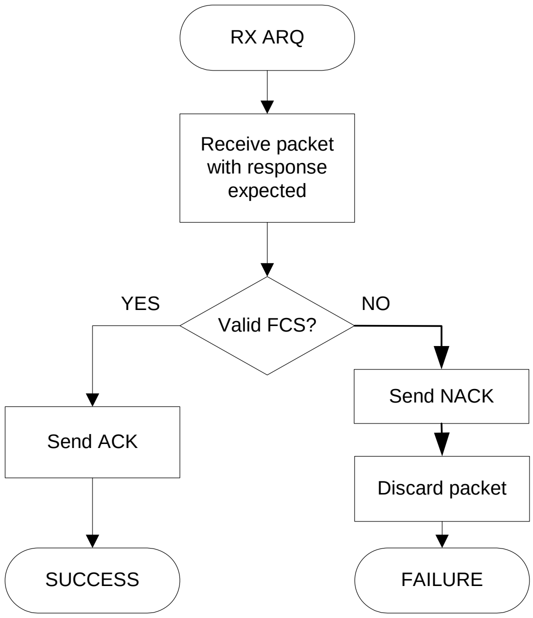


Figure C.5 – Receive ARQ

C.6 Segmentation and reassembly overview

The ITU-T G.9903 PHY layer supports different types of modulation and tone maps. The number of data bytes of the PHY payload can change dynamically based on channel conditions. This requires implementing MAC payload fragmentation on the MAC sublayer. If the size of the MAC payload plus the MAC header is too large to fit within one PSDU, it must be partitioned into smaller segments that can each fit within a PSDU. This process of partitioning the MAC frame into PSDUs is called segmentation and the reverse process is called reassembly. The segmentation may require the addition of padding bytes to the last segment to fit the last PHY frame. The acknowledgement and retransmission occurs independently for the resulting MAC segment. All forms of addressing (unicast and broadcast) are subject to the segmentation.

The segment control field definitions are shown in Table 11.5.

For a packet that requires setting the TMR bit and segmentation, the TMR bit shall be set in the last segment only.

Last segment flag (LSF) shall be set to 1 to indicate the last segment of the MAC packet.

Segment count (SC) shall be set to 0 for the first segment and incremented for each following segment.

Segment length (SL) specifies the length of the MAC payload in bytes for the current segment excluding the MAC header, byte padding and FCS. When security is activated, the MAC payload is constituted of the ciphered payload and the MIC-32.

If segmentation is required to transmit a MAC packet, each resulting MAC frame shall be created as follows:

– The MAC header (MHR) and FCS (MFR) are presented in each segment.

– The first and following segments have the same value of the sequence number assigned for the MAC packet. Only the segment count is incremented for following segments.

– If data encryption is required it must be done before packet segmentation. On the receiver side data decryption is done after packet reassembly.

– All segments except the last one shall set the contention control (CC) bit to inform the receiver that the next PHY frame will be sent in the contention free slot. The last segment clears the contention control bit to allow the normal contention access to the channel.

The segment control fields (see Table 11-5) SL, SC and LSF are used to keep track of segments of the fragmented MAC packet and assembly the whole packet on the receiver side.

Annex D

Modified MAC sublayer data primitives

(This annex forms an integral part of this Recommendation.)

D.1 MCPS-DATA.request

The semantics 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 D.1 specifies the parameters for the MCPS-DATA.request primitive.

Table D.1 – MCPS-DATA.request parameters

Name	Type	Valid range	Description
SrcAddrMode	Integer	0x00-0x03	The source addressing mode for this primitive and subsequent MPDUs. 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 by ITU-T 0x02 = 16-bit short address 0x03 = 64-bit extended address.

Table D.1 – MCPS-DATA.request parameters

Name	Type	Valid range	Description
DstAddrMode	Integer	0x00-0x03	The destination addressing mode for this primitive and subsequent MPDUs. 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 by ITU-T 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. NOTE – PAN identifier value is logically ANDed with 0xFCFF.
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	≤aMaxMACPayload Size	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. Indirect transmission is not supported and bit b2 should always be set to 0. For a non-beacon-enabled PAN, bit b1 shall be set to 0.

Table D.1 – MCPS-DATA.request parameters

Name	Type	Valid range	Description
QualityOf Service	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 (optional).
SecurityLevel	Integer	0x00 and 0x05	The security level to be used as described in clause 11.3.6.
KeyIdMode	Integer	0x01	The mode used to identify the key to be used (see clause 11.3.6).This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0 octets	–	Not used
KeyIndex	Integer	0x00-0x01	The index of the key to be used (see clause 11.3.6).

D.2 MCPS-DATA.indication

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

MCPS-DATA.indication (
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 D.2 – MCPS-DATA.indication parameters

Name	Type	Valid range	Description
SrcAddrMode	Integer	0x00-0x03	The source addressing mode for this primitive and subsequent MPDUs. 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 by ITU-T 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. NOTE – PAN identifier value is logically ANDed with 0xFCFF.
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 MPDUs. 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 by ITU-T 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. NOTE – PAN identifier value is logically ANDed with 0xFCFF.
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	≤aMaxMACPayload Size	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.
msduLink Quality	Integer	0x00-0xFF	The LQI value measured during reception of the message.
DSN	Integer	0x00-0xFF	The DSN of the received frame.
Timestamp	Integer	0x00000000-0xFFFFFFFF	The absolute time in milliseconds at which the frame was received and constructed, decrypted (assuming encryption was valid) (32 bit value).
SecurityLevel	Integer	0x00 and 0x05	The security level to be used as described in clause 11.3.6.

Table D.2 – MCPS-DATA.indication parameters

Name	Type	Valid range	Description
KeyIdMode	Integer	0x01	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 octets	As specified by the KeyIdMode parameter	Not used
KeyIndex	Integer	0x00-0x01	The index of the key to be used (see clause 11.3.6).
QualityOf Service	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 (optional).

Annex E

MAC acknowledgement

(This annex forms an integral part of this Recommendation.)

The present specification does not use the IEEE 802.15.4-2006 MAC acknowledgement frame but specifies positive and negative acknowledgements using the frame control header (see clause 7.4).

The frame control header contains information used by all stations in the network for channel access, as well as PHY receiver information used by the destination. For this reason, the frame control header has specific physical layer encoding and modulation as defined in clause 7.

Only the frame control header will be used as positive (ACK) or negative (NACK) acknowledgement.

The packet originator may request an acknowledgement by setting the delimiter type field of the frame control header (see clause 7.4).

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

If the MAC frame is received without error as determined by the FCS, the receiver shall send an ACK to the originator only if an acknowledgement is requested and the destination address and PANID of the frame match the receiver's device address and PANID.

If the MAC frame is received with errors as determined by the FCS, the receiver may send an NACK to the originator only if an acknowledgement is requested and the destination address and PANID of the frame match the receiver's device address and PANID.

However, if the receiver can determine that the error is caused by collision, it may avoid sending an NACK (no response) to invoke a collision state on the transmitting station. The transmitting station shall infer a collision from the absence of any response to a transmission when a response is expected. In this case the transmitting station shall attempt a retransmission after an EIFS interval.

If a valid NACK is received the transmitting station shall attempt a retransmission using CSMA random back-off.

The receiver will send an NACK to the originator if it is requested and the received MAC frame is corrupted and cannot be recovered by the 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 identifiers and are located in the FCH as follow TM[7:0] = FCS[15:8] and PDC[7:0] = FCS[7:0] (see clause 7.4). The transmitter shall extract the FCS field from the received ACK/NACK and compare it with the FCS of the transmitted packet to determine the validity of the response. If it matches, the ACK/NACK response is accepted otherwise it will be ignored and treated as a collision.

Annex F

Adaptation sublayer service primitives

(This annex forms an integral part of this Recommendation.)

F.1 ADP data service

F.1.1 Overview

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

- ADPD-DATA.request
- ADPD-DATA.confirm
- ADPD-DATA.indication.

F.1.2 ADPD-DATA.request

F.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 F.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 it is concerned with. 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.

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

Name	Type	Valid range	Description
QualityOfService	Integer	0x00-0x02	The requested quality of service (QoS) of the frame to send. Allowed values are: 0x00 = normal priority 0x01 = high priority 0x02 = contention free access (optional).
SecurityEnabled	Boolean	TRUE or FALSE	If TRUE, the frame shall be sent encrypted.

F.1.2.2 When generated

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

F.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. Otherwise, the ADPD constructs a 6LoWPAN frame with the following characteristics depending on the transmission mode.

- In the case of a unicast frame:
  - The mesh addressing header is present as described in clause 5.2 of [IETF RFC 4944], where:
    - V shall be set to 1, to specify that the originator address is a 16-bit network address;
    - F shall 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 shall be present to transport NPDUs which do not fit in an entire IEEE 802.15.4 frame. In this case, clause 5.3 of [IETF 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.
- In the case of a multicast frame:
  - The mesh addressing header is present as described in clause 5.2 of [IETF RFC 4944], where
    - V shall be set to 1, to specify that the originator address is a 16-bit network address;
    - F shall 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 shall be present to transport NPDUs which do not fit in an entire IEEE 802.15.4 frame. In this case, clause 5.3 of [IETF 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 clause 11.4.4 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 the 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 the 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; otherwise it depends on the security policy.

In the 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 the 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; otherwise it depends on the security policy.

If security processing fails for that frame it shall be discarded and an ADPD-DATA.confirm primitive shall 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 shall 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 clause 11.4.4.

F.1.3 ADPD-DATA.confirm

F.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 F.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.

F.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 for failure.

F.1.3.3 Effect on receipt

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

F.1.4 ADPD-DATA.indication

F.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 F.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 received frame was encrypted.

F.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 that has been received.

F.1.4.3 Effect on receipt

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

F.2 ADP management service

F.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
- ADPM-PATH-DISCOVERY.request
- ADPM-PATH-DISCOVERY.confirm.

F.2.2 ADPM-DISCOVERY.request

F.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 F.4 – Parameters of the ADPM-DISCOVERY.request primitive

Name	Type	Valid range	Description
Duration	Integer	0x00-0xFF	The number of seconds the an active scan shall last.

F.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.

F.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.

Upon receiving each beacon frame the MAC layer in the LBD issues an MLME-BEACON-NOTIFY.indication primitive with the PANDescriptor parameters corresponding to the beacon. At the end of scan duration, the adaptation layer generates an ADPM-DISCOVERY.confirm primitive which contains the PANDescriptorList according to the procedure described in clause 11.4.5.2.2.2.

F.2.3 ADPM-DISCOVERY.confirm

F.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 F.5 – Parameters of the ADPM-DISCOVERY.confirm primitive

Name	Type	Valid range	Description
Status	Enum	FAILED, SUCCESS, NO_BEACON	SUCCESS if at least one MLME-BEACON-NOTIFY.indication is received NO_BEACON if no MLME-BEACON-NOTIFY.indication is received In all other cases, FAILED.
PANCount	Integer	0x00-0xFF	The number of networks operating in the POS of the device.
PANDescriptor	List of PAN descriptors	This list contains the PAN descriptors as described in Table F.6. Number of PAN descriptors is specified by PANCount.	The PAN operating in the POS of the device.

Table F.6 – PAN descriptor structure specification

Name	Type	Valid range	Description
PANId	Integer	0x0000-0xFFFF. PAN identifier must be logically ANDed with 0xFCFF	The 16-bit PAN identifier.
LinkQuality	Integer	0x00-0xFF	The 8-bit link quality of LBA. It is used by the associating device to select LBA and PAN.

Table F.6 – PAN descriptor structure specification

Name	Type	Valid range	Description
LBAAddress	Integer	0x0000-0xFFFF	The 16 bit short address of a device in this PAN to be used as the LBA by the associating device.
RC_COORD	Integer	0x00-0xFF	The estimated route cost from LBA to the coordinator. It is used by the associating device to select LBA and PAN.

F.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.

F.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.

F.2.4 ADPM-NETWORK-START.request

F.2.4.1 Semantics of the service primitive

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

The semantics of this primitive are as follows:

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

Table F.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 NOTE – PANId value must be logically ANDed with 0xFCFF.

F.2.4.2 When generated

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

F.2.4.3 Effect on receipt

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

Prior to invoking this primitive, the upper layer of the PAN coordinator shall 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 been formed, the ADP layer shall perform the steps described in clause 11.5.1.

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

F.2.5 ADPM-NETWORK-START.confirm

F.2.5.1 Semantics of the service primitive

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

The semantics of this primitive are as follows:

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

Table F.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.

F.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.

F.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.

F.2.6 ADPM-NETWORK-JOIN.request

F.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 F.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 Annex J.

F.2.6.2 When generated

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

F.2.6.3 Effect on receipt

On receipt of this primitive by a device which has 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 has not already joined, the adaptation sublayer initiates the MAC association procedure ("bootstrap") described in clause 11.4.5.2.2.

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.

F.2.7 ADPM-NETWORK-JOIN.confirm

F.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 F.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	0x0001-0x7FFF, 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. NOTE – PANId value is logically ANDed with 0xFCFF.

F.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.

F.2.7.3 Effect on receipt

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

F.2.8 ADPM-NETWORK-LEAVE.request

This primitive allows a non-coordinator device to remove itself from the network as described in clause 11.4.5.2.2.8. The removal of a device by the coordinator is performed using an ADPM-LBP.request according to the procedure described in clause 11.4.5.2.2.7.

F.2.8.1 Semantics of the service primitive

The semantics of this primitive are as follows:

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

Table F.11 – Parameters of the ADPM-NETWORK-LEAVE.request primitive

Name	Type	Valid range	Description
ExtendedAddress	64-bit address	Any	If NULL, the device removes itself from the network.

F.2.8.2 When generated

The next higher layer of a non-coordinator device generates this primitive to request to leave the network.

F.2.8.3 Effect on receipt

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

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

– If the device is a coordinator or if ExtendedAddress != NULL,
  - Issue ADPM-NETWORK-LEAVE.confirm with INVALID_REQUEST

– Else
  - The device removes itself from the network, using the procedure described in 11.4.5.2.2.8.
  - Issue ADPM-NETWORK-LEAVE.confirm with SUCCESS

F.2.9 ADPM-NETWORK-LEAVE.indication

F.2.9.1 Semantics of the service primitive

This primitive is generated by the ADP layer of a non-coordinator device to inform the upper layer that it has been unregistered from the network by the coordinator. The semantics of this primitive are as follows:

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

Table F.12 – Parameters of the ADPM-NETWORK-LEAVE.indication primitive

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

F.2.9.2 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.

F.2.9.3 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.

F.2.10 ADPM-NETWORK-LEAVE.confirm

F.2.10.1 Semantics of the service primitive

This primitive allows the upper layer to be informed of the status of its previous ADPM-NETWORK-LEAVE.request.

The semantics of this primitive are as follows:

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

Table F.13 – 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.

F.2.10.2 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 clause 11.4.5.2.2.8.

F.2.10.3 Effect on receipt

On receipt, the upper layer is notified of the result of its request.

F.2.11 ADPM-RESET.request

F.2.11.1 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.

F.2.11.2 When generated

This primitive allows a reset of the adaptation sublayer and allows the resetting of the MIB attributes.

F.2.11.3 Effect on receipt

On receipt of this primitive the following steps are performed:

- the adaptation sublayer issues an 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 neighbour tables;
- the adaptation sublayer issues an ADPM-RESET.confirm primitive with the status SUCCESS, or DISABLE_TRX_FAILURE if the MAC reset operation failed.

F.2.12 ADPM-RESET.confirm

F.2.12.1 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 F.14 – 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

F.2.12.2 When generated

This primitive is generated by the ADP layer when a previous ADPM-RESET.request primitive has completed.

F.2.12.3 Effect on receipt

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

F.2.13 ADPM-GET.request

F.2.13.1 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 F.15 – Parameters of the ADPM-GET.request primitive

Name	Type	Valid Range	Description
AttributeId	Integer	See clause 11.4.2	The identifier of the IB attribute to read.
AttributeIndex	Integer	Depends on attribute, see clause 11.4.2	The index within the table of the specified IB attribute to read. This parameter is valid only for IB attributes that are tables.

F.2.13.2 When generated

This primitive is generated by the upper layer to read the value of an attribute from the IB.

F.2.13.3 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.

Otherwise, the adaptation sublayer generates an ADPM-GET.confirm primitive with the status SUCCESS and the value read from the IB in the AttributeValue parameter.

F.2.14 ADPM-GET.confirm

F.2.14.1 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 F.16 – 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 clause 11.4.2	The identifier of the IB attribute read.
AttributeIndex	Integer	Depends on attribute, see clause 11.4.2	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.

F.2.14.2 When generated

This primitive is generated by the adaptation sublayer in response to an ADPM-GET.request primitive.

F.2.14.3 Effect on receipt

On receipt of this primitive the upper layer is informed on the status of its request and eventually gets the desired value.

F.2.15 ADPM-SET.request

F.2.15.1 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 F.17 – Parameters of the ADPM-SET.request primitive

Name	Type	Valid range	Description
AttributeId	Integer	See clause 11.4.2	The identifier of the IB attribute to write.
AttributeIndex	Integer	Depends on attribute, see clause 11.4.2	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.

F.2.15.2 When generated

This primitive is generated by the upper layer to write the value of an attribute in the IB.

F.2.15.3 Effect on receipt

On receipt of this primitive the adaptation sublayer attempts to write 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, and it is not read only but the AttributeValue is out of range, the adaptation layer generates an ADPM-SET.confirm primitive with the status INVALID_PARAMETER. Otherwise, the adaptation layer generates an ADPM-SET.confirm primitive with the status SUCCESS.

F.2.16 ADPM-SET.confirm

F.2.16.1 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 F.18 – 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 clause 11.4.2	The identifier of the IB attribute written
AttributeIndex	Integer	Depends on attribute, see clause 11.4.2	The index within the table of the specified IB attribute written. This parameter is valid only for IB attributes that are tables.

F.2.16.2 When generated

This primitive is generated by the adaptation layer in response to an ADPM-SET.request primitive.

F.2.16.3 Effect on receipt

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

F.2.17 ADPM-NETWORK-STATUS.indication

F.2.17.1 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 F.19 – 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.

F.2.17.2 When generated

This primitive is generated when the adaptation sublayer of a PAN coordinator has received an LBP message from a device on the network indicating that a PAN Id conflict is occurring. See clause 11.5.2 for a complete description of the PAN ID conflict handling mechanism.

In this 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 an MLME-COMM-STATUS.indication.

F.2.17.3 Effect on receipt

On receipt, the upper layer of a PAN coordinator is informed that a PAN Id conflict was detected or that a MAC event occurred.

F.2.18 ADPM-ROUTE-DISCOVERY.request

F.2.18.1 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 F.20 – Parameters of the ADPM-ROUTE-DISCOVERY.request primitive

Name	Type	Valid range	Description
DstAddr	Short address	0x00-0x7FFF	The short unicast destination address of the route discovery.
MaxHops	Integer	0x01-0x0E	This parameter indicates the maximum number of hops allowed for the route discovery.

F.2.18.2 When generated

This primitive is generated by the upper layer of a device to obtain a route to another device.

F.2.18.3 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 clause 11.4.4.2.3.

F.2.19 ADPM-ROUTE-DISCOVERY.confirm

F.2.19.1 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.confirm      (
                                  Status
                                  )
```

Table F.21 – Parameters of the ADPM-ROUTE-DISCOVERY.confirm primitive

Name	Type	Valid range	Description
Status	Status	SUCCESS, INVALID_REQUEST, ROUTE_ERROR	The status of the route discovery.

F.2.19.2 When generated

This primitive is generated by the adaptation layer on completion of a route discovery as described in clause 11.4.4.2.3 and Annex H.

F.2.19.3 Effect on receipt

On receipt 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.

F.2.20 ADPM-PATH-DISCOVERY.request

F.2.20.1 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 F.22 – Parameters of the ADPM-PATH-DISCOVERY.request primitive

Name	Type	Valid range	Description
DstAddr	short address	0-0x7FFF	The short unicast destination address of the path discovery.

F.2.20.2 When generated

This primitive is generated by the upper layer of a device to obtain the path to another device.

F.2.20.3 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 clause 11.4.4.2.4.

F.2.21 ADPM-PATH-DISCOVERY.confirm

F.2.21.1 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-PATH-DISCOVERY.confirm         (
                                    DstAddr,
                                    NSDU
                                    )
```

Table F.23 – Parameters of the ADPM-PATH-DISCOVERY.confirm primitive

Name	Type	Valid range	Description
DstAddr	Short address	0-0x7FFF	The short unicast destination address of the path discovery.
NsduId	integer	N.C	The buffer containing addresses of nodes constituting the path.

F.2.21.2 When generated

This primitive is generated by the adaptation layer on completion of a path discovery as described in clause 11.4.4.2.4 and Annex H.

F.2.21.3 Effect on receipt

On receipt of this primitive the upper layer is informed on the completion of the path discovery.

F.2.22 ADPM-LBP.request

F.2.22.1 Semantics of the service primitive

This primitive allows the upper layer of the client to send the LBP message to the server modem.

The semantics of this primitive are as follows:

```
ADPM-LBP.request   (
                   DstAddrType,
                   DstAddr,
                   NsduLength,
                   NsduId,
                   NsduHandle,
                   NsduType,
```

MaxHops,
DiscoveryRoute,
QualityOfService,
SecurityEnabled
)

Table F.24 – Parameters of the ADPM-LBP.request primitive

Name	Type	Valid range	Description
DstAddrType	Integer	0x02-0x03	The type of destination address contained in the DstAddr parameter. The allowed values are: 0x02 = 2 Bytes address (LBA address) 0x03 = 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
NsduId	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 0x01 = UDP 0x02 = ICMP 0x03 = TCP
MaxHops	Integer	0x01-0x0E	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 requested 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.

F.2.22.2 When generated

This primitive is generated by the LBS to perform the authentication, re-keying and leave procedure.

F.2.22.3 Effect on receipt

On receipt of this primitive the modem sends the coming frame to the destination.

F.2.23 ADPM-LBP.confirm

F.2.23.1 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 F.25 – 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.
NsduHandle	Integer	0x00-0xFF	The handle of the NSDU confirmed by this primitive.

F.2.23.2 When generated

This primitive is generated in response to an ADPM-LBP.request primitive, the status parameter indicates if the request succeeded or the reason for failure.

F.2.23.3 Effect on receipt

On receipt of this primitive the upper layer is notified of the status of a previous ADPM-LBP.request primitive.

F.2.24 ADPM-LBP.indication

F.2.24.1 Semantics of the service primitive

This primitive is used to transfer a received LBP frame from the ADP layer to the upper layer.

The semantics of this primitive are as follows:

```
    ADPM-LBP.indication      (DstAddr,
                          SrcAddr,
                          NsduLength,
                          Nsdu,
                          NsduType,
```

LinkQualityIndicator,

SecurityEnabled

)

Table F.26 – 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.

F.2.24.2 When generated

This primitive is generated by the ADP layer of the client modem when a valid LBP frame whose final destination is the current station has been received.

F.2.24.3 Effect on receipt

On generation of this primitive the upper layer is notified of the arrival of an LBP frame.

F.2.25 ADPM-BUFFER.indication

F.2.25.1 Semantics of the service primitive

This primitive allows the next higher layer to be notified when the modem has reached its capability limit to perform the next frame.

The semantics of this primitive are as follows:

ADPM-BUFFER.indication (

BufferReady

)

Table F.27 – 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

F.2.25.2 When generated

This primitive is generated when the adaptation layer of a modem has reached his limit to perform more Data frame.

F.2.25.3 Effect on receipt

On receipt, the upper layer shall stop the data flow if BufferReady is equal to FALSE and open it if BufferReady is TRUE.

F.3 Behaviour to MAC indications

F.3.1 Overview

This clause describes the behaviour of the adaptation layer in response to an unsolicited indication from the MAC layer.

F.3.2 MCPS-DATA.indication

On receipt of this indication, the adaptation layer shall execute the routing algorithm as described in 9.4.4.

F.3.3 MLME-ASSOCIATE.indication

Nothing shall be done upon receipt of this primitive by the adaptation layer.

F.3.4 MLME-DISASSOCIATE.indication

Nothing shall be done upon receipt of this primitive by the adaptation layer.

F.3.5 MLME-BEACON-NOTIFY.indication

When an MLME-BEACON-NOTIFY.indication is received, and if an ADPM-DISCOVERY.request is currently operating, the adaptation layer shall add the PANId to the PANDescriptorList which will be forwarded to the upper layer in the ADPM-DISCOVERY.confirm primitive.

F.3.6 MLME-GTS.indication

Nothing shall be done upon receipt of this primitive by the adaptation layer.

F.3.7 MLME-ORPHAN.indication

Nothing shall be done upon receipt of this primitive by the adaptation layer.

F.3.8 MLME-COMM-STATUS.indication

On receipt of this primitive, the adaptation layer shall 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 ":".

F.3.9 MLME-SYNC-LOSS.indication

The adaptation layer shall respond to the reception of this primitive as described in clause 11.5.2.

Annex G

Device Starting Sequence of messages

(This annex forms an integral part of this Recommendation.)

Each device shall start with an adpDeviceType attribute of Not_Device, Not_Server (see Table 11-22) 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 device or server mode and optionally set the PIB parameters to configure it.

c) If the equipment is a device it shall perform the following steps:

- discovery procedure by invoking the ADPM-DISCOVERY.request;
- if there is a device or a server in its POS, it shall then invoke the ADPM-NETWORK-JOIN.request to perform the bootstrapping procedure.

d) Otherwise (if the equipment is a server) it shall perform the following steps:

- discovery procedure by invoking the ADPM-DISCOVERY.request;
- if there is no device in the server's POS, it shall invoke the ADPM-NETWORK-START to start a network; otherwise, it 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 as described in clause 11.5.1. The procedures and decisions associated with this behaviour are implementation specific.

Equipment cannot send or receive data packets unless they have joined the network.

Annex H

6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD)

(This annex forms an integral part of this Recommendation.)

NOTE 1 – This annex is copied from IETF draft-daniel-6lowpan-load-adhoc-routing-03: 6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD). Edited by K. Kim, S. Daniel, G. Montenegro, S. Yoo, N. Kushalnager. June 19, 2007.

NOTE 2 – In this annex, the term "TBD" refers to items left for further study.

Network Working Group

Internet-Draft

Intended status: Standards Track

K. Kim, Ed.	S. Daniel Park, Ed.
picosNet Corp/Ajou Univ.	SAMSUNG Electronics
G. Montenegro	S. Yoo
Microsoft Corporation	Ajou University
N. Kushalnagar	
Intel Corp	

June 19, 2007

Status of this Memo

Abstract

6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD) is intended for use by IEEE 802.15.4 devices in a 6LoWPAN. It is a simplified on-demand routing protocol based on AODV.

H.1 Introduction

The [IEEE 802.15.4] standard targets low power personal area networks. The "IPv6 over IEEE 802.15.4" document [I-D.montenegro-lowpan-ipv6-over-802.15.4] defines basic functionality required to carry IPv6 packets over IEEE 802.15.4 networks (including an adaptation layer, header compression, etc.).

Likewise, the functionality required for packet delivery in IEEE 802.15.4 meshes is defined, as mesh topologies are expected to be common in LoWPAN networks. However, neither the IEEE 802.15.4 standard nor the "IPv6 over IEEE 802.15.4" specification provide any information as to how such a mesh topology could be obtained and maintained.

The 6LoWPAN Ad hoc Routing Protocol (LOAD) is a simplified on-demand routing protocol based on AODV [RFC 3561] for 6LoWPAN. Besides the main AODV specification [RFC 3561], several efforts aim at simplifications of the protocol, as in the AODVjr proposal [AODVjr] or the TinyAODV implementation [TinyAODV]. Similarly, DyMO allows for minimalist implementation leaving non-essential functionality as optional [I-D.ietf-manet-dymo]. LOAD enables multihop routing between IEEE 802.15.4 devices to establish and maintain routing routes in 6LoWPAN.

This document defines the message formats, the data structures, and the operations of LOAD.

H.2 Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119].

H.3 Overview

This section describes the distinctive features of LOAD compared to AODV. LOAD is defined to be operating on top of the adaptation layer instead of the transport layer. That is, it creates a mesh network topology underneath and unbeknownst to IPv6 network layer. IPv6 sees a 6LoWPAN as a single link. This is similar to how other technologies regularly create complex structures underneath IP (e.g., ethernet spanning tree bridges, token ring source routing, ATM, etc.). LOAD control packets use the encapsulation defined in [I-D.montenegro-lowpan-ipv6-over-802.15.4]. All LOAD control packets shall use the prot_type value TBD (suggested value of 4).

LOAD assumes the use of either one of the two different addresses for routing: the EUI-64 address and the 16 bit short address of the 6LoWPAN device.

LOAD makes use of broadcast in its route discovery. It does so in order to propagate the Route Request (RREQ) messages. In this specification, such broadcast packets are obtained by setting the PAN id to the broadcast PAN (0xffff) and by setting the destination address to the broadcast short address (0xffff).

LOAD doesn't use the destination sequence number in order to reduce the size of the control messages and simplify the route discovery process. For ensuring loop freedom, only the destination of a route SHOULD generate a RREP in reply. The intermediate nodes SHOULD not respond with a RREP. By the same reason, LOAD does not use the "Gratuitous RREP".

LOAD MAY use the local repair for a link break during a data delivery. In a local repair, only the destination generates a RREP in reply because of no use of the destination sequence number.

If a local repair fails, LOAD MAY generate a Route Error (RERR) message towards the originator of the data delivery to notify that the destination is no longer reachable by way of the broken link. The format of RERR is simplified to include only one unreachable destination while the RERR of AODV MAY include multiple ones.

LOAD does not use the "precursor list" of AODV to simplify the routing table structure. Notice that AODV uses the precursors for forwarding RERR messages in the event of detection of the loss of the next hop link. In LOAD, RERR is forwarded only to the originator of the failed data delivery, thus no requiring to use the precursor list.

LOAD MAY use the route cost, which is the accumulated link cost from the originator to the destination, as a metric of routing. For this, LOAD utilizes the Link Quality Indicator (LQI) of the 6LoWPAN PHY layer in the routing decision in addition to the hop distance. There are many ways to include LQI in the routing metric. The approach taken by LOAD avoids a route which contains weak links whose LQI is below certain threshold value (i.e., WEAK_LQI_VALUE).

LOAD SHOULD utilize the acknowledged transmission option at the 6LoWPAN MAC layer for keeping track of the connectivity of a route. LOAD uses neither the passive acknowledgements nor the HELLO messages of AODV.

The basic operations of LOAD are route discovery, managing data structures and maintaining local connections. For these operations, LOAD maintains the following two tables: the routing table and the route request table. The routing table stores route information such as destination, next hop node, and status. The route request table stores the temporary route information used in the route discovery process.

There are two different types of 6LoWPAN devices: the reduced function device(RFD) and the full function device (FFD). LOAD SHOULD utilize only FFD for mesh routing. Thus, A FFD SHOULD implement the operations of LOAD and maintain the data structures of LOAD.

H.4 Terminology

This section defines the terminology of LOAD that is not defined in [RFC 3753] and [RFC 3561].

Destination: A node to which data packets are to be transmitted. Same as "destination node".

forward route: A route set up to send data packets from the originator to its destination.

link cost: The link Quality (LQ) between a node and its neighbor node.

link quality indicator (LQI): A mechanism to measure the link quality (LQ) in IEEE 802.15.4 PHY layer. It measures LQ by receiving the signal energy level. A high LQ value implies the good quality of communication (i.e., low link cost).

weak link: A link of which the LQI falls below WEAK_LQI_VALUE.

originator: A node that initiates a route discovery process. Same as "originating node".

route cost: An accumulated link cost as a LOAD control message (RREQ or RREP) passes through the nodes on the route.

reverse route: A route set up to forward a RREP back to the originator from the destination. Same as "reverse route" in [RFC 3561].

H.5 Data Structures

A FFD in 6LoWPAN SHOULD maintain a routing table and a route request table. This section describes the tables and the message formats.

H.5.1 Routing Table Entry

The routing table of LOAD includes the following fields:

destination address — The 16 bit short or EUI-64 link layer address of the final destination of a route.

next hop address — The 16 bit short or EUI-64 link layer addresses of the next hop node to the destination.

Status — The status of a route. It includes the following states: VALID, INVALID, ROUTE_DISCOVERY, etc.

life time — The valid time in milliseconds before the expiration or the deletion of a route.

H.5.2 Route Request Table Entry

Route request table is used for discovering routes. It stores the following route request information until a route is discovered.

route request ID — A sequence number uniquely identifying the particular RREQ when taken in conjunction with the originator.

originator address — The 16 bit short or EUI-64 link layer address of the node which originates a RREQ.

reverse route address — The 16 bit short or EUI-64 link layer address of the next hop node on the reverse route to the originator.

forward route cost — The accumulated link cost along the forward route from the originator to the current node through which a RREQ is forwarded.

reverse route cost — The accumulated link cost along the reverse route from the final destination to the current node through which a RREP is forwarded.

valid time — The time of the expiration or deletion of a route in milliseconds.

H.5.3 Message Format

H.5.3.1 Route Request (RREQ)

<table>
<tr><td></td><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>3</td><td></td></tr>
<tr><td>Bits</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td></td><td colspan="8">Type</td><td>R</td><td>D</td><td>O</td><td colspan="5">Reserved</td><td colspan="4">CT</td><td colspan="4">WL</td><td colspan="8">RREQ ID</td></tr>
<tr><td></td><td colspan="8">RC</td><td colspan="24">Link layer Destination Address</td></tr>
<tr><td></td><td colspan="32">Link layer Originator Address</td></tr>
</table>

Figure H.1 – RREQ message format

The RREQ message format is shown in Figure H.1 and contains the following fields:

Type — 1 for indicating a RREQ message.

CT — Type of route cost. The followings are the current route cost types known:

0: Hop count while avoiding weak links

1-0xf: TBD

WL — The total number of weak links on the routing path from the originator to the sender of the RREQ.

R — 1 Local Repair.

D — 1 for the 16 bit address of the destination,

0 for the EUI-64 address of the destination.

O — 1 for the 16 bit address of the originator,

0 for the EUI-64 address of the originator.

RC(Route cost) — The accumulated link cost of the reverse route from the originator to the sender of the RREQ. The type of link cost is specified by CT.

RREQ ID — A sequence number uniquely identifying the particular RREQ when taken in conjunction with the originator.

Reserved — 0; ignored on reception.

Link layer Destination Address — The 16 bit short or EUI-64 link layer address of the destination for which a route is supplied.

Link layer Originator Address — The 16 bit short or EUI-64 link layer address of the node which originated the Route Request.

H.5.3.2 Route Reply (RREP)

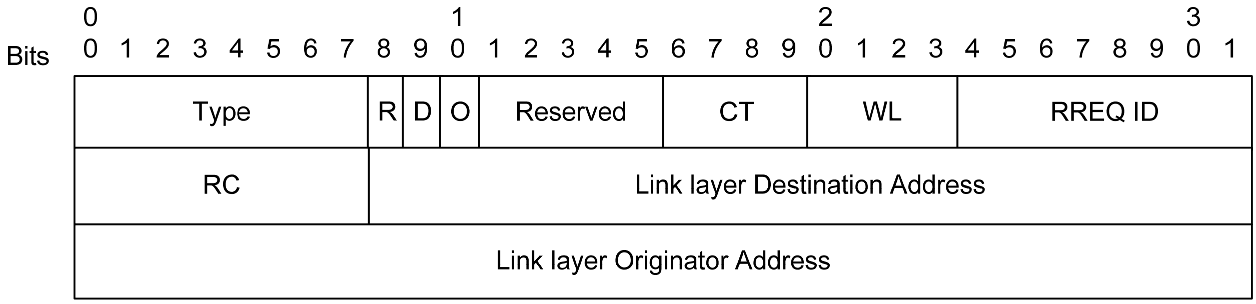


Figure H.2 – RREP message format

The RREP message format is shown in Figure H.2 and contains the following fields:

Type 2 for indicating a RREP message.

CT Type of route cost. The followings are the current route cost types known:

0: Hop count while avoiding weak links

1-0xf: TBD

WL The total number of weak links on the routing path from the originator of the RREP to the sender of the RREP.

R 1 Local Repair.

D 1 for the 16 bit address of the destination,

0 for the EUI-64 address of the destination.

O 1 for the 16 bit address of the originator,

0 for the EUI-64 address of the originator.

Reserved 0; ignored on reception.

RC(Route cost) The accumulated link cost of the route from the originator of the RREP to the sender of the RREP. The type of link cost is specified by CT.

RREQ ID A sequence number uniquely identifying the particular RREQ when taken in conjunction with the originator.

Link layer Destination Address The 16 bit short or EUI-64 link layer address of the destination for which a route is supplied.

Link layer Originator Address The 16 bit short or EUI-64 link layer address of the node which originated the Route Request.

H.5.3.3 Route Error (RERR)

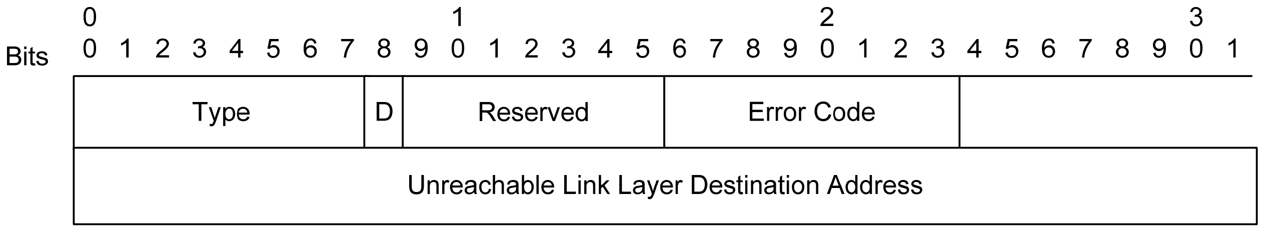


Figure H.3 – RERR message format

The RERR message format is shown in Figure H.3 and contains the following fields:

Type	3 for indicating a RERR message.
D	1 for the 16 bit address of the destination, 0 for the EUI-64 address of the destination.
Reserved	0; ignored on reception.
Error Code	Numeric value for describing error. 0x00 = No available route 0x01 = Low battery 0x02 = routing cost not supported 0x03 – 0xff = reserved (TBD)
Unreachable Link Layer Destination Address	The 16 bit short or EUI-64 link layer address of the final destination that has become unreachable due to a link break.

H.6 Operation

H.6.1 Generating Route Request

The basic operations of LOAD include route discovery, managing data structures and maintaining local connections. A node maintains the following two tables for routing: the routing table and the routing request table.

During the discovery period, an originator, a node that requests a route discovery, generates a Route Request (RREQ) message with the RREQ ID which was incremented by one from the previous RREQ ID value.

A node SHOULD NOT originate more than RREQ_RATELIMIT RREQs per second. After broadcasting a RREQ, a node waits for a RREP. If a route is not discovered within NET_TRAVERSAL_TIME milliseconds, the node MAY try again the discovery process a maximum of RREQ_RETRIES times.

H.6.2 Processing and Forwarding Route Request

Upon receiving a RREQ, an intermediate FFD node tries to find the entry of the same originator address and RREQ ID pair in the route request table. If the entry is found, the node just discards the RREQ. Otherwise, the node creates a reverse route to the originator in the routing table and a RREQ entry in the route request table. It then checks whether the link through which the RREQ is received is a weak link or not. If the link is a weak link, the node adds 1 to the WL field of the RREQ. Then, the node forwards the RREQ.

H.6.3 Generating Route Reply

When the destination receives a RREQ, it tries to find the entry of the same originator address and RREQ ID pair in the route request table. If the entry is found, the destination compares the route cost of the RREQ with the forward route cost of the entry. If the cost of the RREQ is better than (i.e., less than) that of the entry, the destination updates the reverse route to the originator in the routing table and generates a RREP in reply. If the cost of the RREQ is not less than that of the entry, the destination just discards the RREQ.

If the CT field of the RREQ is 0 (i.e., hop count while avoiding weak links), the route cost becomes a tuple of (WL, RC) and is ordered lexicographically. That is, the route cost (WL, RC) is said to be better (or smaller) than or equal to (WL', RC') if the following condition holds.

(WL,RC) <= (WL',RC') if and only if WL < WL', or WL == WL' and RC <= RC'

H.6.4 Receiving and Forwarding Route Reply

Upon receiving a RREP, an intermediate node checks whether the link through which the RREP is received is a weak link or not. If the link is a weak link, the node add 1 to the WL field of the RREP.

The node then checks whether it has a route entry for the destination of the RREP (i.e., the originator of the corresponding RREQ). If it does not have the route entry, it just discards the RREP. Otherwise, it also checks for the existence of the corresponding RREQ entry (which has the same RREQ ID and originator address pair as that of the RREP) in the route request table. If there is no such entry, then it just discards the RREP.

If there is such an entry and the entry has worse reverse route cost (i.e., higher value) than the route cost of the RREP, the node updates the entry with the information of the RREP and forwards it to the previous hop node toward the destination of the RREP. If the entry has better reverse route cost (i.e., lower value) than that of the RREP, the node just discards the RREP.

If the CT field of the RREP is 0 (i.e., hop count while avoiding weak links), the route cost becomes a tuple of (WL, RC) and is ordered lexicographically.

During the delivery of the RREP to the originator, the route cost value of the RREP is accumulated on the reverse route from the destination to the originator.

H.6.5 Local Repair and Route Error (RERR) Messages

If a link break occurs or a device fails during the delivery of data packets, the upstream node of the link break MAY repair the route locally. To repair a route, the node disseminates a RREQ with the originator address set to its own address and the destination address set to the data packet's destination address. In this case, the 'R flag' of the RREQ is set to 1. The data packet is buffered during the route discovery period. If the destination node receives the RREQ for a route repair, it responds with a RREP of which the 'R flag' is also set to 1.

If the repairing node cannot receive a RREP from the final destination until the end of the route discovery period, it unicasts a RERR with an error code that indicates the reason of the repair failure to the originator. A repairing node SHOULD NOT generate more than RERR_RATELIMIT RERRs per second. Then, the buffered data packet is discarded. If the originator that sends a data packet receives the RERR, it MAY try to reinitiate route discovery.

When the repairing node receives a RREP from the destination during the route discovery period, it updates the routing table entry information from the RREP. Then the node transmits the buffered data packet to the destination through the new route.

H.7 Configuration Parameters

This section describes the default values for some important parameters associated with LOAD operations.

Parameter Name	Value
--------------------	-------------
NET_TRAVERSAL_TIME	TBD
RREQ_RETRIES	3
RREQ_RATELIMIT	2
RERR_RATELIMIT	2
WEAK_LQI_VALUE	8

H.8 IANA Consideration

This document needs an additional IANA registry for the prot_type value that indicates the LOAD format.

H.9 Security Considerations

The security considerations of the [RFC 3561] are applicable to this document. As described in the charter of the 6lowpan, e.g., LOAD will also try to reuse existing security considerations related to Ad hoc routing protocols. Further considerations will be studied in the next version.

H.10 Acknowledgments

Thanks to the authors of RFC 3753 and RFC 3561, as parts of this document are patterned after theirs. Thanks to Nandakishore Kushalnagar, Byeong-Hee Roh, Myung-ho Jung, Dae-hong Son, and Minho Lee for their useful discussions and supports for writing this document.

H.11 References

H.11.1 Normative Reference

…………………………

…………………………

H.11.2 Informative Reference

…………………………

…………………………

Annex J

Commissioning in 6LoWPAN

(This annex forms an integral part of this Recommendation.)

NOTE 1 – This annex is copied from IETF draft-6lowpan-commissioning-02: Commissioning in 6LoWPAN. Edited by K. Kim, S. Shams, S. Yoo, S. Park, G. Mulligan. July 15, 2008.

NOTE 2 – In this annex, the term "TBD" refers to items left for further study.

Network Working Group

Internet-Draft

Intended status: Standards Track

Expires: January 16, 2009

K. Kim, Ed.
S. Shams
picosNet Corp/Ajou Univ.
S. Park, Ed.
SAMSUNG Electronics
G. Mulligan

S. Yoo
Ajou University

July 15, 2008

Status of this Memo

Abstract

The commissioning process defines the startup procedure executed by any 6LoWPAN device. This document defines the startup procedure that should be followed by a 6LoWPAN device in any open or secured network.

J.1 Introduction

6LoWPAN is a low-power wireless personal area network(LoWPAN) which is comprised of the IEEE 802.15.4-2006 standard [IEEE 802.15.4] devices. One of the design goal for 6LoWPAN architecture is to ensure minimum human intervention during provisioning a sensor device in a PAN. However, a

6LoWPAN device requires a set of pre-deployed information, called LoWPAN Information Base(LIB), to find the right PAN, to successfully join with the PAN, and to establish communication within the PAN. A device needs specific procedure, what we named as a Bootstrapping protocol for 6LoWPAN device, to collect those information from LoWPAN Bootstrapping Server (LBS) and to start communication in a PAN. This procedure needs to be well defined for interoperability of devices from different vendors. This procedure involves extracting LIB, security credentials, becoming part of existing network, obtaining 16-bit short address, and IP settings.

J.2 Terminology

Active Scan: An active scan is used by a device to locate all coordinators transmitting beacon frames within its personal operating space, which is provided by IEEE 802.15.4. It requests other devices to transmit the beacon frame.

association: An IEEE 802.15.4 device can be assigned a dynamic 16 bit short address during an association operation with a neighbor device (or router) which is also called as the parent device. After getting the short address, a device can communicate with its parent or child by using only the assigned short address.

coordinator: A full-function device (FFD) which is the principal controller of a 6LoWPAN. It is also called as PAN coordinator. It MAY initiate the synchronization of the entire 6LoWPAN by transmitting beacons.

ED Scan: An ED scan allows a device to obtain a measure of the peak energy in each requested channel, which is provided by IEEE 802.15.4.

Full Function Device (FFD): A device implementing the complete protocol set of IEEE 802.15.4. It is capable of operating as a router (multi-hop packet forwarding) for its associated neighbors.

Neighbor Table: A table which has the information of neighbor devices in a personal operating space.

LoWPAN Bootstrapping Information Base (LIB): A set of pre-deployed information that is necessary for a particular 6LoWPAN device to find the desired PAN and to successfully join with the PAN. We categorize this information into two groups; PAN Specific Information (PSI), which is the same for every device in a PAN, (for example, PAN ID), and Device Specific Information(DSI), which is specific for each particular node (for example short address).

PSI: PAN Specific Information Inside the LIB, a portion of information, called PSI, is the same for every device in the target PAN. For example, PAN_ID, PAN_Type, etc.

DSI: Device Specific Information Inside the LIB, other than PSI, there is some information that may vary from device to device. For example, Role_of_Device, Short_Addr, etc.

LoWPAN BootStrapping Device (LBD): LBD is a device that is needed to be deployed in the target network. LBD is assumed to have no priori information about the 6LoWPAN within which it is going to join. The only information it has is the EUI-64 address and a "Join key" (in case of secured PAN).

LoWPAN BootStrapping Server (LBS): An entity that contains LIB of each device to be bootstrapped. It indexes this information with the EUI-64 address of each 6LoWPAN device. LBS has two modules in it; Network management & Account Module (NAM) and Authentication Module (AM). NAM keeps track of the LIB of each device indexed by EUI-64 address whereas AM participates in authentication process on behalf of LBD using LBD's 'Authentication credentials'. Based on the 'LBP Message', LBS verifies LBD with the help of Authentication server (in case of secured PAN) and sends ACCEPT message with necessary information otherwise it sends DECLINE message. In the case of secured PAN, LBS initiates authentication mechanism issuing Authentication request into appropriate format that is acceptable by particular authentication server. Any challenge or reply message from the Authentication server is encapsulated in the 'LIB message' by LBS and is sent back to the LBD through LBA.

LoWPAN BootStrapping Agent (LBA): A FFD that has already joined in the PAN and thus, it is already a member of the PAN. It is also a neighbor of a new LBD, and thus it helps the bootstrapping LBD by receiving LBP message from LBD and forwarding it to LBS.

Open 6LoWPAN: An open 6LoWPAN is a PAN where any device is welcomed.

Close 6LoWPAN: A close 6LoWPAN is a PAN where only pre-defined set of devices are allowed to join based on their EUI-64 address. This account is managed by LBS. If close 6LoWPAN is secured, it is called secured 6LoWPAN.

Secured 6LoWPAN: Secured 6LoWPAN is a Close 6LoWPAN that also maintains secured message exchange in the PAN.

PAN Id: The 16 bit 6LoWPAN identifier which is administratively assigned to a 6LoWPAN and is unique within the PAN.

Passive Scan: A passive scan, like an active scan, is used by an FFD to locate all coordinators transmitting beacon frames within its personal operating space, which is provided by IEEE 802.15.4. The difference is that the passive scan is a receive-only operation and does not request the beacon frame.

Personal Operating Space (POS): The area within the reception range of the wireless transmission of a IEEE 802.15.4 packet.

Reduced Function Device (RFD): A IEEE 802.15.4 device of 6LoWPAN which does not have the functionality of the router. That is, it can not forward IPv6 packets to the next hop device. It can only be the end device of 6LoWPAN.

Short Address: A 16 bit address dynamically assigned to a device from the PAN.

J.2.1 Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119].

J.3 Bootstrapping

Bootstrapping is defined as collecting LIB from LBS, obtaining security credentials (optional), associating with the right PAN, obtaining 16-bit short address (optional), and constructing IPv6 address using IPv6 prefix. Specifically, this includes the process of starting the network, associating with other nodes, obtaining the unique IPv6 address, and constructing security credentials for 6LoWPAN.

J.3.1 Resetting the device

After the device is started, it first performs a MAC layer reset.

J.3.2 Scanning through channels

During this phase, functions supported by 802.15.4 are used for scanning channels. Appendix (A.1) shows the scanning process in 802.15.4. For getting the information of other devices within POS, the device should perform scan. The device can use either an active scan or a passive scan. During scanning procedure, the device receive beacon frames from other devices.

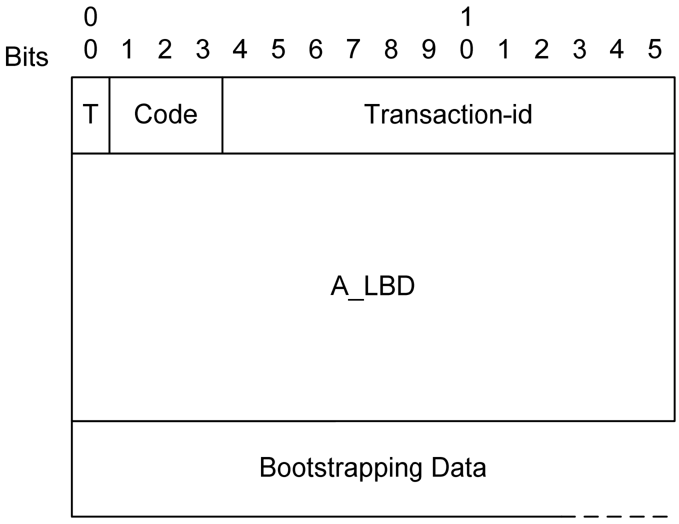
J.3.3 LoWPAN BootStrapping Mechanism

This protocol defines mechanism to extract LIB from currently unknown LBS and also defines a message format for LIB message exchange. In this protocol, LBD exchanges LBP message with LBS through its one hop neighbor LBA. So, at the beginning of LBP, it needs to find an LBA using 'LBA discovery phase' that is described in section 3.3.2.

J.3.3.1 LoWPAN BootStrapping Protocol message format

In this section we define a message format which is necessary for LBP.

J.3.3.1.1 LBP message



T Type of message

It defines message type. value '0' represents 'Message from LBD' and '1' represents 'Message to LBD'.

Code:

000, 1xx: Reserved.

001 ACCEPTED. Authentication of LBD has been accepted.

010 CHALLENGE. It indicates that authentication process has not been finished. Authentication server has sent some challenge that has to be replied by LBD.

011 DECLINE. In the case of unsecured 6LoWPAN, LBS may send this code to indicate that LBD's EUI-64 address is not allowed to join the PAN. In case of secured 6LoWPAN, LBS may send this code to indicate that LBD's EUI-64 address is not allowed to join the PAN or the authentication of the LBD is failed.

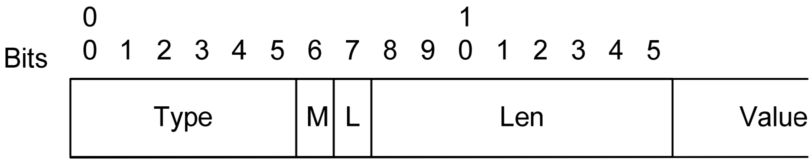
Seq Sequence Number

Seq identifies the number of messages transmitted by LBD. Corresponding incoming message from LBS should also have the same Seq.

A_LBD Address of Bootstrapping Device (LBD)

64-bit EUI-64 address of LBD.

Bootstrapping Data Format of bootstrapping data is given below.



Type: 6-bit represents the ID of the attribute in LIB if 'L' bit is set. Otherwise, this field defines particular authentication type.

A list of authentication mechanism and their corresponding 'Type' is TBD.

M: Type of the Attribute

This field defines the type of the attribute in LIB; whether it is PAN Specific Information (PSI) or Device Specific Information (DSI). 1 represents PSI and 0 represents DSI.

Len: 8-bit represents the length of the value in octet.

Value: This field represents the corresponding data of the type.

J.3.4 LoWPAN Bootstrapping Information Base

One of the important goal of LBP is to receive a set of information from LBS by a joining LBD. This information comprises of PSI and DSI. Following table shows attribute name, attribute ID (attr_ID), purpose of the attribute and type of it.

Attribute Name........Attribute ID......Attribute Description PSI/DSI

Attribute Name	Attr_ID	Type	Attribute Description
PAN_ID	1	P	This is the network identification for the default network
PAN_type	2	P	Secured/closed/open
Address_of_LBS	3	P	Address of the LBS. 0x0000 in case of no LBS. For example in open 6LoWPAN.
Join_Time	4	P	It specifies the time when this node should start trying to join the target PAN.
Role_of_Device	5	D	Agent/No_Agent
Allow_LBA_To_Send_PSI	6	P	This attribute allows any SF to provide GI to CD after getting the positive reply from LBS.
Short_Addr	7	D	16-bit address for new device which is unique inside the PAN
Short_Addr_Distribution_ Mechanism	8	P	Its Value is either 0 or 1 representing central or distributed respectively. If it is central, short address is provided by LBS itself otherwise assigning short address is
Other_Device_Specific _Info	15	D	Using this attribute, a device and LBS can exchange any types of data or security key required by the device.

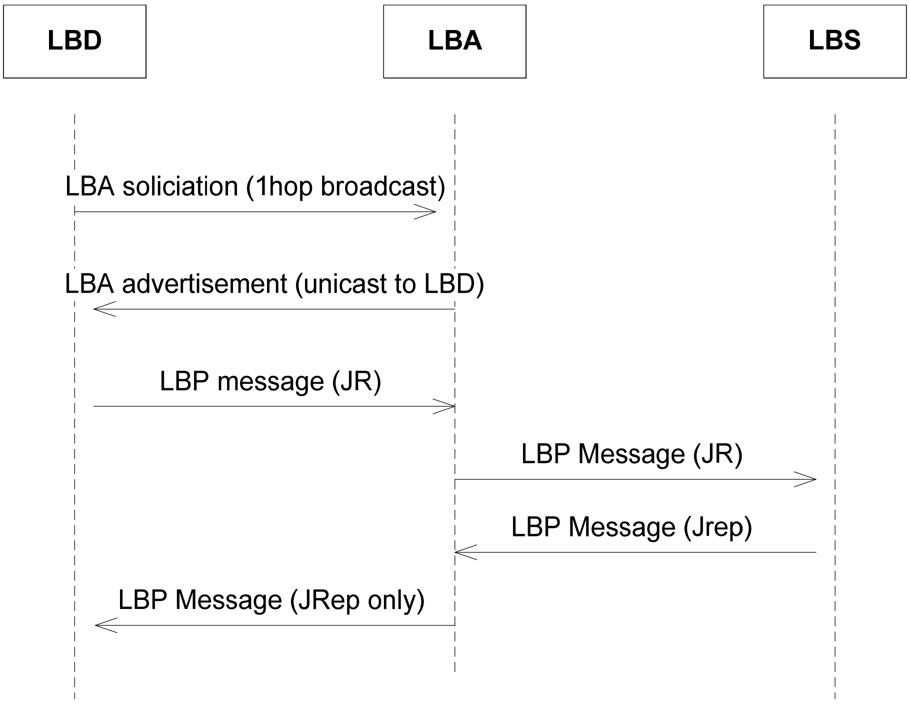
J.3.4.1 LBA discovering phase

LBD has to send LBP message to the LBS server under the support of a LBA. To find the LBA, it broadcasts a LBA solicitation message within its one hop neighbors and waits for a LBA advertisement. Any device capable of being LBS/LBA replies to the broadcast specifying its capability as LBS/LBA. If there is any LBS in its neighbor, LBD selects that LBS otherwise it selects one of the LBAs.

J.3.4.2 LoWPAN Bootstrapping Protocol (LBP)

LBD sends LBP message to LBA, as it doesn't know the address or path to the LBS of the target PAN. LBA forwards the LBP message to LBS on behalf of LBD. LBS replies with one or multiple LBP messages destined to LBA as LBD still is not part of the network. If the network is secured 6LoWPAN and the LBD is an authentic node, we assume that LBD has necessary pre-deployed keys and the knowledge of the authentication mechanism necessary to authenticate in target PAN. In this case, LBD sends necessary information in the 'bootstrapping data' field so that LBS can initiate the authentication process using that 'authentication credentials'. LBS converts the LBP message into appropriate authentication request for the particular authentication server and sends it. A reply/challenge from the authentication server, for example EAP authenticator or AAA server, is encapsulated in LBP message's 'bootstrapping data' field and is sent back to the LBD through LBA. LBA also keeps track of the successful authentication, failed authentication and incomplete conversation of the authentication process, and maintains a 'black list' of malicious devices to avoid repeated attack. Detecting malicious device based on those 3 information and marking that node as 'Black listed' belongs to the scope of security policy and out of the scope of this draft.

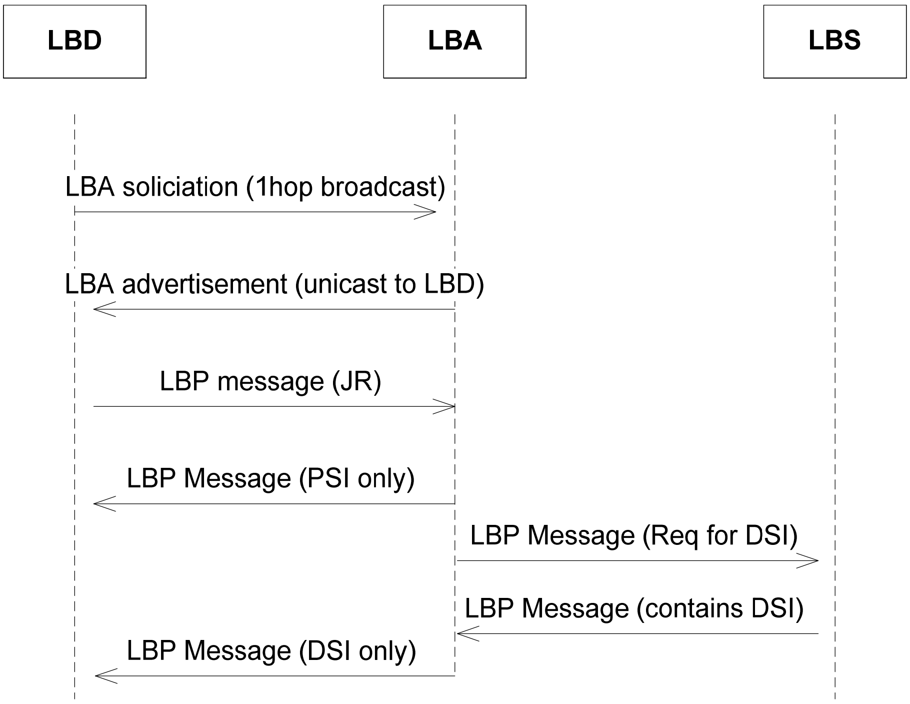
Following figure shows a simple example of Bootstrapping mechanism.



JR = Join Request, JRep = Join Reply

J.3.4.3 Bootstrapping in open 6LoWPAN:

An open 6LoWPAN network, usually welcomes any willing LBD. In this case, it doesn't need to wait for reply from LBS. Instead, LBA can provide GI from its own LIB and can forward LIB request to LBS simultaneously.


LBD
LBA
LBS
LBA soliciation (1hop broadcast)
LBA advertisement (unicast to LBD)
LBP message (JR)
LBP Message (PSI only)
LBP Message (Req for DSI)
LBP Message (contains DSI)
LBP Message (DSI only)

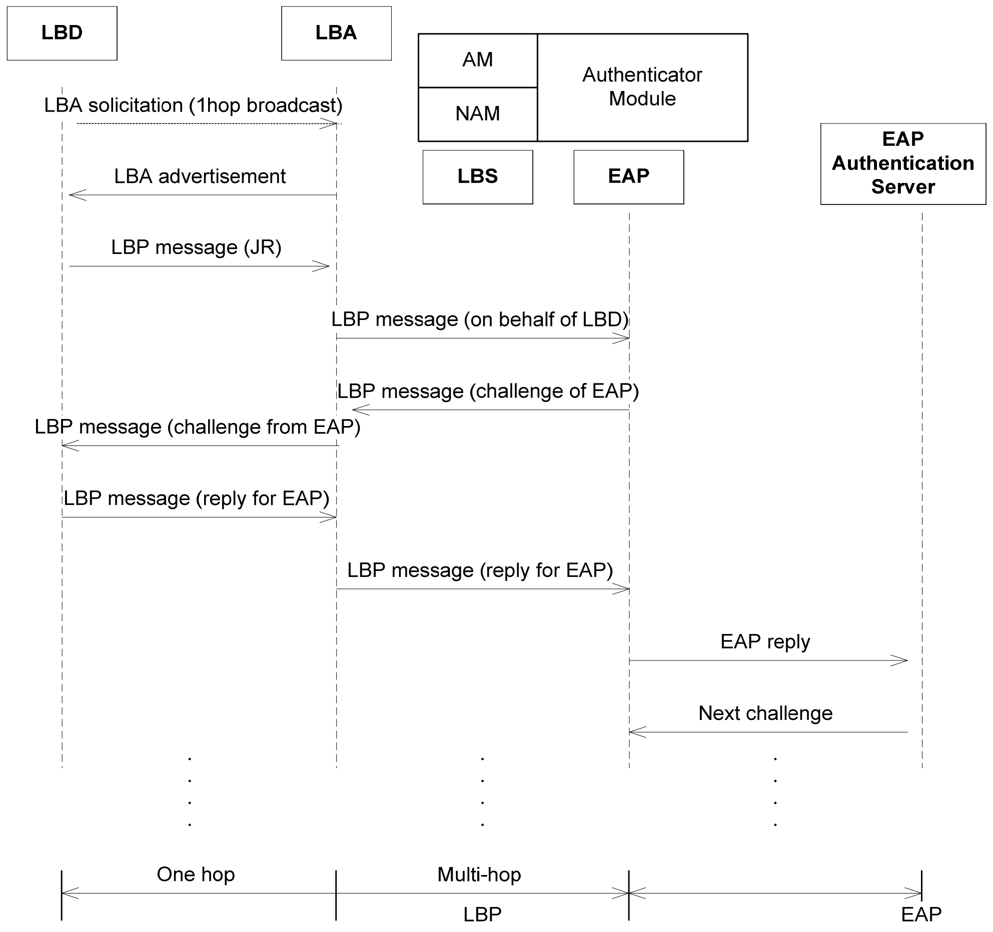
JR = Join Request.

J.3.4.4 LBP in secured 6LoWPAN

In secured 6LoWPAN, LBD must has to exchange authentication credentials using its join key. Apart from requesting network resources, in the case of secured network, this process may need to exchange several encrypted message between LBD and authentication server. LBA and LBS serves as 'secured tunnel' for authentication message exchange process. Both LBA and LBS keep the account of the last LIB request/reply processed by themselves.

Example: LBP with EAP

The following figure shows how LBP with other authentication protocol like EAP works. At first LBD broadcasts a LIB request (1 hop) to LBA. LBA already has a secured route to LBP so it just unicasts the LIB request to LBS. LBS sends an EAP packet prepared with LBD's authentication credentials and sends it to authenticator. it is also possible that LBS entity and authenticator entity resides on a single system. As discussed earlier, LBS serves as translator between LBP and EAP message exchange in this authentication process and finally when AM indicates the success of authentication, it sends all network resources along with the ACCEPTED code. In the case of failure in authentication process, DECLINE code is sent to LBD.


LBD
LBA
AM
NAM
Authenticator Module
LBS
EAP
EAP Authentication Server
LBA solicitation (1hop broadcast)
LBA advertisement
LBP message (JR)
LBP message (on behalf of LBD)
LBP message (challenge of EAP)
LBP message (challenge from EAP)
LBP message (reply for EAP)
LBP message (reply for EAP)
EAP reply
Next challenge
One hop
Multi-hop
LBP
EAP

J.3.4.5 Role of Entities in LBP

Role of LoWPAN Bootstrapping Device (LBD):

- It selects LBA using LBA discovery phase.
- If it doesn't find any LBA, it gives up after waiting for certain amount of time.
- if it receives any LBP message with code "CHALLENGE", it must send another LBP message containing the appropriate value against the challenge/query in the bootstrapping data field.
- It MUST increment seq for every new LBP message. For retransmission seq should remain same.

Role of LoWPAN Bootstrapping Agent (LBA):

When LBA receives LBP message from LBD.

1. If the LBD is already in the Black List, discard
2. If the LBD is new, and 6LoWPAN is open network,
   a) Send 'LBP message' with ACCEPTED along with all PSI from its own LIB.
   b) If there is any LBS in the PAN, Forward the 'LBP message' to LBS for DSI.
3. If the LBD is old, and 6LoWPAN is open network
   a) If it matches with the last seq no. send the last reply.
   b) Otherwise discard.
4. If the LBD is new, and 6LoWPAN is secured network
   a) forward the LBP message to LBS
5. If the LBD is old, and 6LoWPAN is secured network
   a) If it matches with the last seq no. send the previously saved last LBP message 'for LBD'.
   b) If the LBP has completed, discard.
   c) If the LBP is 'CHALLENGE' and new seq is right next of the last one, forward the message to LBS.

When LBA receives LBP message from LBS (for LBD)

– if it is ACCEPTED and 16-bit short address is the responsibility of LBA, it calculates and appends the 16-bit short address with the LIB reply.

– Otherwise, if it is ACCEPTED, DECLINED or CHALLENGE, forward it to the corresponding LBD.

– If it is not ACCEPTED or DECLINED, delete previously saved LBP message and save this LBP message.

– If it is DECLINED, based on the security policy, mark it as 'Blacklisted'.

– If there is no activity in some of the flow (LBD-LBS pair), mark the LBD and based on the security policy include it in 'Black list'.

Role of LoWPAN Bootstrapping Server (LBS):

In the case of open 6LoWPAN

– If the LBD is 'valid' that means its EUI-64 is in accepted list or not in the rejection list, it sends ACCEPTED code and necessary DSI and 16-bit short address(if the address should be assign centrally).

In the case of secured 6LoWPAN

– AM of LBS determines authentication server for particular EUI address and sends authentication mechanism initiation with the authentication credentials to that authentication server.

– when it gets reply from authentication server, if it is success, it prepares a success reply if it is failure, it prepares a failure reply f it is challenge/query, it prepares processing reply for LBDand sends to LBA.

– When AM module receives success from authentication server, it informs success to NAM module and sends the success response to NAM.NAM then, sends DSI along with the response in LBP message.

J.3.5 Assigning the short address

During LBP procedure, LBD may set a short address either by itself or receiving the address from the PAN. The short address must be unique in a PAN and may be given by a centralized or distributed way.

One of the approach to distribute the short address among the LBDs is centralized fashion where a centralized entity (e.g., LBS) assigns 16-bit short address for LBD. Allocation of short address MAY be based on First-Available-Address-First or randomly chosen one or using any other algorithm.

Distributed approach is another way to assign 16-bit short address to LBDs. In this approach, LBA assigns short address to the joining device, LBD. A hierarchical addressing scheme could be used by LBA in this purpose. Following figure describes the address calculation scheme. This scheme requires one parameter MC, the maximum number of addresses a LBA can assign. If the present LBD is the first children, then it gets the short address by following formula,

FC = MC * AP + 1

where FC is the LBD address, and AP is the address of the LBA.

If LBD is not the first child of this LBA, it receives the address which is next to the last address assigned by that LBA.

For example, if LBA(1) assigned address 6 to its last LBD, it assigns address 7 to its next LBD.

MC = 4

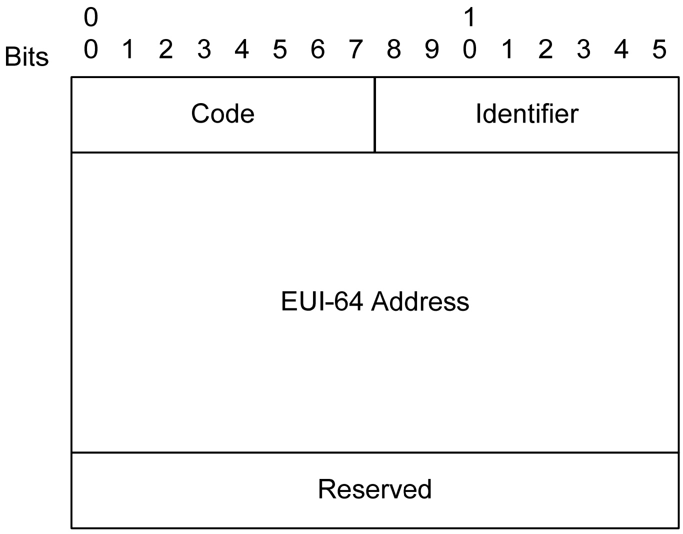
```
                (0)  <= Coordinator
               // \\
               / | | \
               / / \ \
           (1) (2) (3) (4) <= Routers
           // \\ ......... // \\
          / / \ \        / / \ \
  (5) (6) (7)(8)..(17)(18)(19)(20)
                // \\
               / / \ \
 ...........(69) (70)(71) (72)........
```

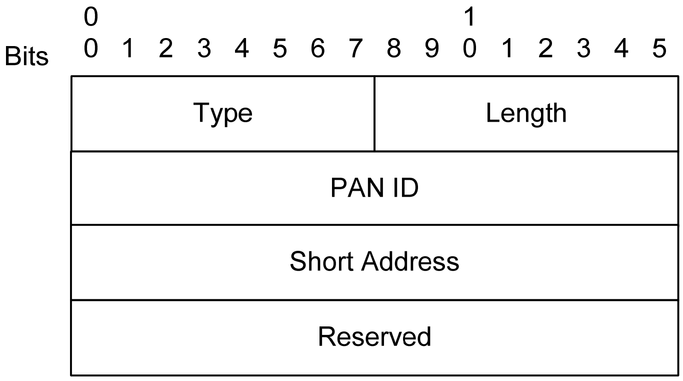
Fig. The assignment scheme of the short address

J.3.6 Obtaining IPv6 address

The IPv6 interface identifier of a device can be obtained as described in Section 6 of [RFC 4944]. After having a unique IPv6 interface identifier, the device begins to obtain an IPv6 address prefix. The IPv6 address prefix for a particular 6LoWPAN is stored by the IPv6 router in the 6LoWPAN. ICMPv6 is used to share these parameters. Routers in 6LoWPAN are supposed to broadcast Router Advertisements(RA) messages periodically. The RA message must contain the prefix option which can be used in the 6LoWPAN. Devices wish to obtain IPv6 address prefix may wait for an RA message until RA_WAIT_TIME elapsed. After that, if no RA message is received, they may broadcast Router Solicitation RS) message for requesting the RA message.

The RS and RA messages can have additional option fields as described in [RFC 4861]. Source/Target link-layer address option field should contain the EUI-64 address or the combined address with PAN ID and 16bit short address of the source or target device as below. The RS and RA messages can have additional option fields as described in [RFC 4861]. Source/Target link-layer address option field should contain the EUI-64 address or the combined address with PAN ID and 16bit short address of the source or target device as below.





Source/Target Link-layer Address option field

Multiple IPv6 routers could form a single or multiple 6lowpan(s). If there are multiple routers in a 6LoWPAN, the device should consider which one is to be selected as a default router. One possible way of selection is to compare the hop counts travelled of the RA message of each router. The detailed algorithm for the selection is TBD.

J.3.7 Configuration Parameters

This section gives default values for some important parameters associated with the 6LoWPAN commissioning protocol. A particular node may wish to change certain of the parameters.

Parameter Name	Value
---------------------	-----
CHANNEL_LIST	0xFFFF800
SCAN_DURATION	3
SUPERFRAME_ORDER	15
BEACON_ORDER	15
START_RETRY_TIME	1 000 msec
JOIN_RETRY_TIME	4 000 msec
ASSOCIATION_RETRY_TIME	4 000 msec

J.4 IANA Consideration

TBD.

J.5 Security Considerations

IEEE 802.15.4 devices is required to support AES link-layer security. MAC layer also provides all keying material necessary to provide the security services. It isn't defined, however, when security shall be used especially combining with Bootstrapping. After the device start and join the network, security services such as key management and device authentication should be done automatically. Detailed algorithm for security on Bootstrapping is TBD.

J.6 Contributors

Thanks to the contribution from MD. Aminul Haque Chowdhury (Ajou Univ) and Chae-Seong Lim (Ajou Univ) for the review and useful discussion for writing this document.

J.7 Acknowledgments

Thanks to Hamid Mukhtar (PicosNet/Ajou Univ), Jae-ho Lee (NIA), and Dong-Gyu Nam (NIA) for their useful discussion and support for writing this document.

J.8 References

J.8.1 Normative References

[RFC 2119] Bradner, S., "*Key words for use in RFCs to Indicate Requirement Levels*", BCP 14, RFC 2119, March 1997.

[RFC 4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "*Neighbor Discovery for IP version 6 (IPv6)*", RFC 4861, September 2007.

[RFC 4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "*Transmission of IPv6 Packets over IEEE 802.15.4 Networks*", RFC 4944, September 2007.

[IEEE 802.15.4] IEEE Computer Society, "IEEE Std. 802.15.4-2006".

J.8.2 Informative References

[RFC 3513] Hinden, R. and S. Deering, "*Internet Protocol Version 6 (IPv6) Addressing Architecture*", RFC 3513, April 2003.

Annex K

Regional requirements for Japan

(This annex forms an integral part of this Recommendation.)

For further study.

Appendix I

Examples on encoding and decoding

(This appendix does not form an integral part of this Recommendation.)

I.1 Example for data encoding

Suppose we have a 40-byte MAC packet to send in DQPSK mode (2 bits per symbol) with 25 carriers available (due to notching and/or tone-mapping).

The size of the data at the interleaver input is equal to inter_input_size = (((40×8) + (16×8)) + 6) × 2 = 908 bits (Reed-Solomon adds 16 bytes, the convolutional encoder adds 6 bits and multiplies the size by 2).

The minimal interleaver buffer size:

– We have 25 carriers, so m = 25.

– We have n = FL×4×bits_per_symbol, so

  FL = ceiling(inter_input_size/(m × 4 × bits_per_symbols))

  = ceiling(908/(25×4×2)) = ceiling(4,54) = 5 and n = 40.

As m=25 and n=40, the matrix can "store" 1000 bits and the data is 908 bits long, so 92 bits of padding must be added. Those 92 bits of padding are split between byte padding and bit padding, the byte padding being maximized (with the constraint that the input is in bytes). So the upper layer shall add floor (92/2/8) = 5 bytes of padding before the data enters the scrambler, and the 12 remaining bits of bit padding shall be added by the PHY layer at the interleaver input.

I.2 Example for data decoding

When decoding a frame, we need to compute the amount of bit padding to process the frame. The FCH contains the following information (decoding the example in above clause):

– FL = 5

– DQPSK modulation (2 bits per symbol)

– 25 carriers used (tone-map + notching information)

So, the interleaver buffer can hold 25 × (4 × FL × 2) = 1 000 bits.

In these 1 000 bits:

– 16×8×2 bits were added by Reed-Solomon.

– 12 bits were added by the convolutional encoder.

– The remaining 732 bits are a mix of data and padding:

  – The data part is equal to floor (732/2/8) bytes = 45 bytes.

  – The bit padding is equal to 732 – (data_size × 8 × 2) bits = 12 bits.

In the 45 bytes of data, 5 bytes of byte padding are removed by the MAC layer using the "Segment length" header information.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	**Transmission systems and media, digital systems and networks**
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2013