Work Package:	SSWG
Activities:	
Type of document:	Deliverable
Date:	31/01/06

IST Integrated Project No 507667

Partners:	Ascom / Dimat / Dresden University of Technology / DS2 / EDF / Endesa / EPFL / Iberdrola / Iberinco / Main.net / PPC / Robotiker / Schneider Electric / Spidcom / Telvent / Unika / Yitran
Responsible:	EDF, DS2
Circulation:	☒ **Public** EC ☐ **Confidential** ☐ **Restricted**
File name:	OPERA_Specification_Part_1_Technology_v1.0.doc **Version:** 1.0
Title:	OPERA SPECIFICATION - Part 1: TECHNOLOGY

OPERA TECHNOLOGY SPECIFICATION – PART 1

COPYRIGHT

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

This document is the technical specification of the OPERA technology.

1.1 Scope

This document specifies a physical layer (PHY), medium access control (MAC), LLC layer, convergence layer and coexistence protocol for data transmission over the electrical powerlines for Access Networks.

1.2 Overview

The purpose of this document is to specify a powerline data transmission system based on Orthogonal Frequency Division Multiplexing (OFDM) for providing Access services.

Specifically this standard,

- Describes a PHY capable of achieving rates in excess of 200 Mbps
- Describes a Master-Slave MAC optimised for the Access powerline environment
- Describes the QoS mechanisms available to support bandwidth and latency guarantees
- Describes the security procedures used to provide data privacy over the powerline medium
- Describes coexistence mechanisms between OPERA and In-home systems

The description is written from the transmitter perspective to ensure interoperability between devices and allow different implementations. Some minimal requirements about the receiver behaviour are also given.

1.3 References

The following standards contain provisions which, through references in this text, constitute normative provisions of this specification. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent revisions of the standards listed below.

- OPERA Specification: Part 2.

- FIPS PUB 46-3: 1999. Data Encryption Standard (DES)
- FIPS PUB 81: 1980. DES modes of operation
- ANSI X9.42: 2003. Agreement of Symmetric Keys Using Discrete Logarithm Cryptography
- FIPS PUB 186-2: 2000. Digital Signature Standard (DSS)
- IEEE 802.3 Standards for Local and Metropolitan Area Networks – Residential Ethernet
- IEEE 802.1p Standards for Local and Metropolitan Area Networks - Traffic Class Expediting and Dynamic Multicast Filtering
- IEEE 802.1D Standards for Local and Metropolitan Area Networks – MAC bridges
- IEEE 802.1q Standards for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks
- IEEE 802.1w Standards for Local and Metropolitan Area Networks - Rapid Reconfiguration of Spanning Tree

1.4 Document Conventions

This document is divided into sections and appendices. The document body (all sections) is normative (except for italicised text); each appendix is identified as in its title as normative or informative.

Text formatted in italics is not part of the specification. It is for clarification only.

Binary numbers are indicated by the prefix 0b followed by the binary digits, for example, 0b0101. Hexadecimal numbers are indicated by the prefix 0x.

Formal requirements are indicated with 'shall' in the main body of this document.

Options are indicated with 'may' in the main body of this document. If an option is incorporated in an implementation, it shall be done as specified in this document.

$\lceil \cdot \rceil$ denotes rounding to the closest higher or equal integer

$\lfloor \cdot \rfloor$ denotes rounding to the closest lower or equal integer

1.5 Definitions

Active slave	An Active slave regularly gets transmission opportunities via the reception of data tokens from its master.
Bridge port	Ports of the bridging block. There are three types of bridge ports: PLC ports, non-PLC ports and the management port. PLC ports are the bridge ports that are directly mapped onto the completed entries of the Port Solver Table. Non-PLC ports are any other external ports of the node (e.g. Ethernet, USB, WIFI,…). The management port is used for the exchange of management messages between the bridging block and the layer management.
Bridging block	A functional block which is placed above the convergence layer. This block makes uses of standard 802.1D bridging to perform a layer-2 interconnection between the bridge ports of a node.
Broadcast	A transmission that is addressed to all the units of a layer-2 Local Area Network. It corresponds to the transmission of an Ethernet frame with an Ethernet Broadcast Destination address. It might be carried over the broadcast port or can be transmitted consecutively over unicast ports.
Broadcast port	Specific LLC port which value is 127. Transmission over the broadcast port makes use of the HURTO modulation.
Burst	A MAC Service Data Unit passed to the MAC layer by the LLC layer. A Burst includes one or several complete or fragmented packets destined to the same port.
Channel Estimation Frame	A Channel Estimation MPDU which does not contain any data payload. A Channel Estimation Frame starts with a Token Announce followed by a specific sequence of symbols used for channel estimation. It is not terminated by a token.
CPE	A PLC unit which only behaves as a slave node.
Data Payload	Burst payload which is adaptive modulated or HURTO modulated, depending on the channel conditions.
Delimiter	A delimiter is a 22 byte block which is HURTO modulated over one single symbol. There are three types of delimiters: the Burst Header (LLC), the Token Announce and the Token (MAC).
FDR	A PLC unit which combines a HE and either a CPE or a TDR.

Frame	An MPDU passed by the MAC layer to the PHY layer. It can be a regular MPDU or a Channel Estimation MPDU.
HE	A PLC unit which only behaves as a master node. It is the root of a PLC cell. Contrarily to a TDR, the master part of a HE is not linked to a slave part from which the right to communicate depends.
Idle slave	An Idle slave does not get any transmission opportunities from its master. However, it is regularly polled by its master. (see polling).
Master	A node responsible for controlling access of its slaves to the network.
Message	Management Protocol Data Units generated by the Layer Management block. Management messages are IEEE 802.3 SNAP encapsulated and provided to the bridging block before being subsequently transmitted onto the bridge ports. Management messages are either unicast or multicast. Multicast management messages are transmitted over the broadcast port using HURTO modulation. Unicast management messages are transmitted over the unicast port using the adequate mapping (adaptive or HURTO).
Mode	A mode is defined by a symbol type, a carrier center frequency and an optional power mask
MPDU	Mac Protocol Data Unit delivered by the MAC layer to the PHY layer. The MAC layer generates two kinds of MPDUs: regular MPDUs and Channel Estimation MPDUs. A Channel Estimation MPDU does not contain any data payload (see Channel Estimation Frame definition). A regular MPDU can contain zero, one or several bursts addressed to one or several ports. A regular MPDU starts with a Token Announce delimiter and is terminated by a Token delimiter. There are 6 types of regular MPDUs: data, polling, access, access reply, silence and non-returnable. The type of the regular MPDU corresponds to the type of the Token terminating the regular MPDU.
Multicast	A transmission that is addressed to a group of units of a LAN segment. It is making use of an Ethernet Multicast Destination address. It might be carried over port 127 using HURTO mode or can be transmitted consecutively over unicast ports.
Node	A logical element which implements this specification.
Packet	An LLC Service Data Unit passed by the Convergence layer to the LLC layer. Each Ethernet frame received from the Convergence layer is converted into one single packet.

PHY-signal	PHY-signals are specific signals used as PPDU headers. PHY-signals include the SOT, SYNC and CREF signals.
PLC cell	A set of units composed by a HE and the units under the direct or indirect control of that HE. A PLC cell is operating under the same symbol type and the same carrier center frequency.
PLC subcell	A set of units composed by a master and the units under the direct control of that master
PLC sub-tree	A set of units composed by a TDR and the units under the direct or indirect control of that TDR.
Polling	Process under which a master regularly checks the status of its Idle slaves. There are two types of polling identified by the type of the polling token. By using the Alive polling token, the master can detect if a slave is not connected anymore (Unregistered slave). By using the Active polling token, the master can detect if a slave is requiring transmission opportunities (Active slave).
Port	LLC 7-bit address identifier. Communication between node A and B requires opening a Local Port on node A and B. The Local Port on node A is defined as the Remote Port on node B and vice-versa. Transmission is performed onto a Local Port whereas reception is performed from a Remote Port. From a receiver standpoint, a Remote Port is not unique: a remote node is uniquely identified by its Remote Port and its MAC address. Port entries are managed via the Port Solver Protocol and the Announce Messages. Port 127 is used as the broadcast port. Authorized port values are 9 to 126.
Port Solver Table	Addressing table used by the LLC layer. It contains entries with the association between MAC addresses of the distant nodes and ports. The entries are complete when both the local and remote ports are assigned or incomplete when only the local port is assigned and the remote port is set to 0xFF.
Registered slave	A slave whose access to the network has been authorized via the Access Protocol. A Registered slave can be Active or Alive.
REP	Refers to either TDR or FDR
Slave	A node for which transmission opportunities are controlled by its master. These transmission opportunities are specified as Validity Periods carried in some specific tokens delivered by the master. Slaves which are part of a TDR can delegate these transmission opportunities to their associated internal master (see TDR definition).

SOT	A specific PHY-signal which starts all PPDUs. It is also used as a positive acknowledgement of a polling request.
Spatial reuse	A feature which enables simultaneous node transmissions within non interfering PLC sub-trees of a single PLC cell. This feature relies on the non-returnable token and the Cluster Discovery Protocol
OFDM Symbol	Transmitted signal for that portion of time when the modulating amplitude and phase state is held constant on each of the equally-spaced subcarriers in the signal.
Symbol type	One from three different kinds of OFDM symbol, according to its duration.
TDR	A PLC unit made of both an internal master part and a slave part which never operate at the same time. A TDR behaves either as a master or as a slave at different time periods. Contrarily to a HE, the master part of a TDR is not a permanent master which has permanent control of its transmission opportunities. The right to communicate of the master part of a TDR depends on the Validity Period assigned to its associated slave part.
Token	The token is the MAC delimiter which terminates a regular MPDU. There are six types of tokens: data token is used to propose transmission opportunities to a distant node, access/access reply tokens are used to discover Unregistered slaves, polling token is used for the slave states polling, non-returnable token is used for allocating simultaneous transmission opportunities to several slaves (spatial reuse), silence token is used for sending a regular MPDU and keeping control of the medium after the transmission of this MPDU.
Unicast	A transmission that is addressed to a unique recipient.
Unregistered slave	An Unregistered slave is not managed by any master. A master can detect such slaves via the exchange of an access frame and an access reply frame.
Visibility	From a node to another node, when the first node is able to demodulate the token announce sent by the second node.
Word	A data unit of 32-bits

1.6 Abbreviations and acronyms

4D	Four Dimensional
AAA	Authorization – Authentication – Accounting
ABLP	Adaptive Bit-Loading Protocol
ABR	Available Bit Rate
ACK	ACKnowledge
AFE	Analogue Front End
AGC	Automatic Gain Control
ANSI	American National Standards Institute
APL	Aggregated Payload Length
ARQ	Automatic Repeat Request
ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
BE	Best Effort
BER	Bit Error Rate
BH	Burst Header
BNDA	Border Node Designation Acknowledge
BNDP	Border Node Designation Protocol
BPC	Bits Per subCarrier
BPDU	Bridge Protocol Data Unit
BPS	Bits Per Symbol
CAC	Connection Admission Control
CBR	Constant Bit Rate
CCITT	Comité Consultatif International Téléphonique et Télégraphique

CDP	Cluster Discovery Protocol
CLPDU	Convergence Layer Protocol Data Unit
CP	Cyclic Prefix
CPE	Customer Premises Equipment
CRC	Cyclic Redundancy Check
CSM	Class of Service Monitor
CTS	Clear To Send
CW	CodeWord
DAC	Digital to Analogue Converter
DES	Data Encryption Standard
DS	DownStream
DSP	Digital Signal Processing
EMC	Electromagnetic Compatibility
FCMP	Fair Congestion Management Policy
FEC	Forward Error Correction
FIPS	Federal Information Processing Standards
FFT	Fast Fourier Transform
FTP	File Transfer Protocol
FW	Firmware
HE	Head End
HURTO	High Ultra Reliable Transmission for Ofdm
IEEE	Institute of Electrical and Electronics Engineers
ICI	Inter Carrier Interference

IDFT	Inverse Discrete Fourier Transform
IDP	Interference Detection Packet
IQ	In-phase and Quadrature
ISI	Inter Symbol Interference
ISO	International Organization for Standardization
LLC	Logical Link Control
LP	Local Port
LPDU	LLC Protocol Data Unit
LSB	Least Significant Byte
LSDU	LLC Service Data Unit
LV	Low Voltage
MAC	Media Access Control
MID	Master Identification
MPDU	MAC Protocol Data Unit
MSB	Most Significant Byte
MSDU	MAC Service Data Unit
MV	Medium Voltage
OFDM	Orthogonal Frequency Division Multiplexing
OPERA	Open PLC European Research Alliance
OPERACP	OPERA Communication Protocol
OSI	Open Systems Interconnection
OUI	Organizationally Unique Identifier
OVLAN	Optimized VLAN

PAR	Peak to Average Ratio
PBPS	Physical Bits Per Symbol
PCMP	Priority Congestion Management Policy
PDU	Protocol Data Unit
PHY	PHYsical layer
PLC	PowerLine Communications
PN	Pseudo Noise
PPDU	Physical Protocol Data Unit
PSD	Power Spectral Density
PST	Port Solver Table
QoS	Quality of Service
REP	Repeater
RP	Remote Port
RS	Reed Solomon
RTS	Request To Send
RX	Reception
SNAP	SubNetwork Access Protocol
SNR	Signal to Noise Ratio
SLA	Service Level Agreement
SOT	Start of Transmission
STP	Spanning Tree Protocol
TA	Token Announce
TCM	Trellis Coded Modulation

TCP	Transfer Control Protocol
TDD	Time Division Duplexing
TDMA	Time Division Multiple Access
TDR	Time Division Repeater
TO	TimeOut
TX	Transmission
UDP	User Datagram Protocol
UPA	Universal Powerline Association
US	UpStream
VBR	Variable Bit Rate
VLAN	Virtual Local Area Network
VoIP	Voice over IP

2 GENERAL DESCRIPTION

2.1 General description of the architecture

An access PLC network consists of a number of user terminals (CPEs) that transmit/receive traffic in a shared medium to/from a centralized station (HE). If the signal is too attenuated to reach all CPEs from the same HE, repeaters (REP) can be inserted in the network in order to retransmit the signal and thus increase the coverage. Repeaters can be either TDR or FDR. From this description, the type of topologies to be found in PLC access networks are tree-like topologies, like the one depicted in Figure 1, where a central node, called a HE, concentrates all of the upstream and downstream traffic.

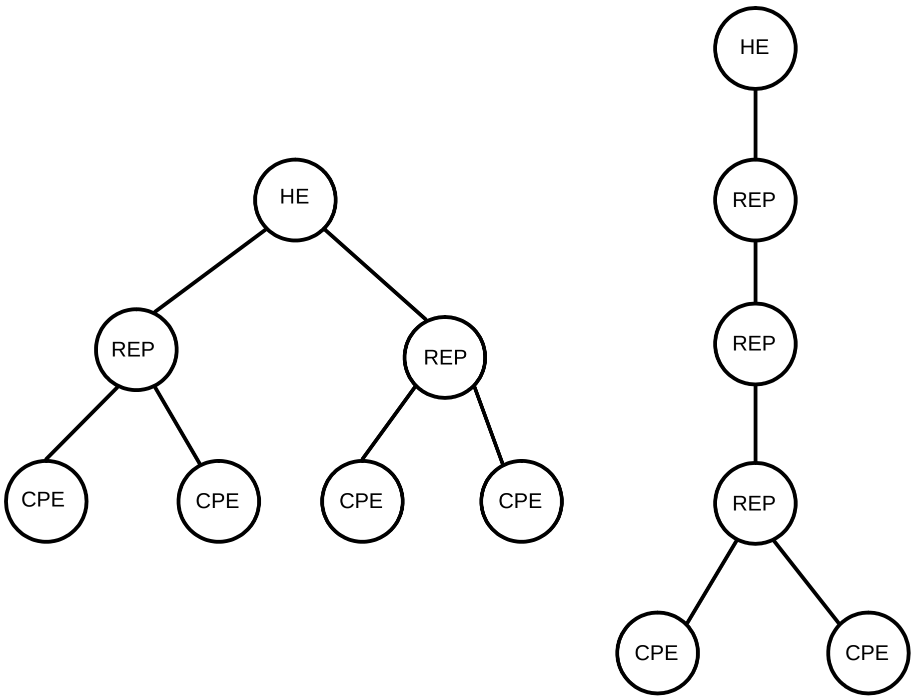

Figure 1 Typical Access Scenario

2.2 Node description

A PLC network can be made either of:

- one PLC cell,
- several PLC cells if they are interconnected via Frequency Division repeaters (FDR). FDR are made of a combination of one HE and one CPE or TDR operating in different modes.

In an access PLC network, there is usually one HE, placed in the MV/LV substation, that concentrates the upstream and downstream traffic from/to all the CPEs of all the connected PLC cells.

A PLC cell is composed by a HE and the set of units (CPEs or TDRs) under the direct or indirect control of that HE operating under the same symbol type and the same carrier frequency. Itis composed of:

1. One HE
2. From zero to several TDRs
3. One (or several) CPE(s)

2.2.1 HE

The HE is the central node that controls the entire PLC cell. It assigns resources to all nodes of the PLC cell through the use of the token, according to the QoS requirements of the flows circulating on the PLC cell. The HE will always be the master of any node directly connected to it.

2.2.2 Time Division Repeaters

A TDR is used to increase the coverage in areas too far from the cell's HE. TDRs are connected to the HE or to other TDRs that act as their master node. TDRs share the channel allocated to them by their master node and distribute it among their slave nodes according to each one's QoS profile. The TDR will be the slave of the HE or of another TDR, and will be the master of its slaves.

2.2.3 CPE

CPEs are PLC units installed in the customer's household. A CPE must subscribe to the network before being able to access the channel. Subscribing to the network means selecting a master node that assigns channel access time. In order to subscribe to the network, a validation process is run that acknowledges that the CPE is valid. After being accepted into the network, the CPE automatically downloads a file (auto-configuration process) detailing the parameters to use, such as the user QoS profile and other configuration parameters. The CPE is always a slave.

2.3 Network reference model

This specification defines PHY, MAC, LLC and Convergence Layer functionality as well as minimal bridging requirements to be supported by all compliant implementations. It also defines layer management functions. Finally, it also contains specifications for Security and Coexistence.

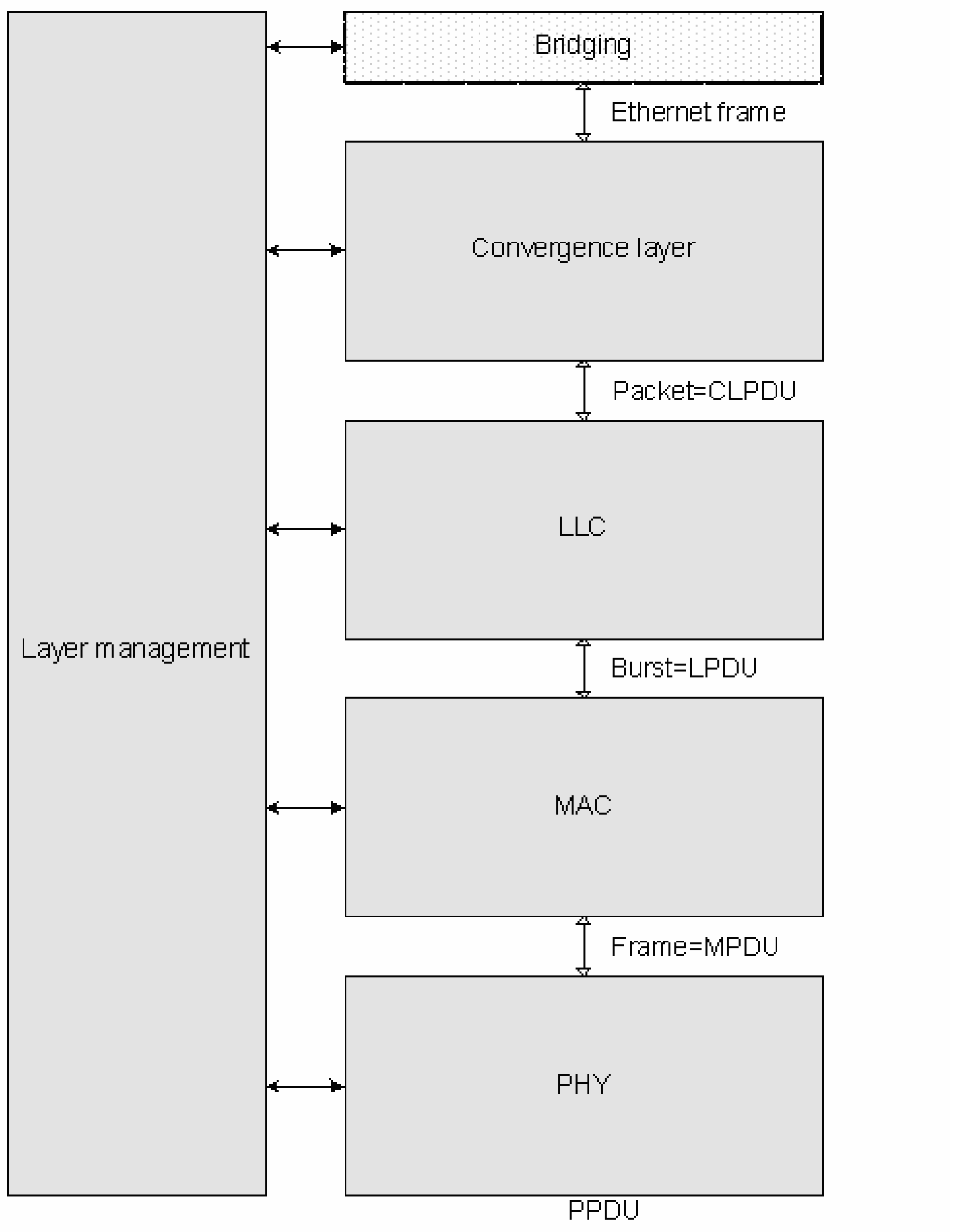

Figure 2 System Reference Model

This document specifies Packet Management, LLC, MAC and PHY behaviour. End-to-end services are provided by protocol layers not specified in this document.

At an application level the system appears as a black box between information packet interfaces and the power line.

The Bridging block is not described in this technology specification. It is mandatory and described in the System Specification (OPERA Part 2). From a bridging standpoint, the layer management block is connected to the bridging block via a management port. The Convergence Layer is seen as a set of PLC ports. The specific PLC requirements are described in 7.

The Convergence layer performs the Ethernet Frame to Packet conversion. This function performs the conversion between Ethernet II/802.3 frames (802.1 p/q tagged or untagged) and 32-bit aligned packets. On the transmission path, the converter also sets the priority and the OVLAN tag, if needed. The Broadcast/Multicast handling is also done at this level.

The LLC layer provides peer convergence layers with the ability to exchange LLC service data units (LSDUs=packets). Given a maximum size constraint on the LPDU payload provided by the layer management, the LLC segments and/or groups packets into LPDU payloads (burst payloads) to be transferred over the same transmission port. When several packets are present within a burst payload, inter-packets headers are appended. In relation with the layer management, the LLC also deals with encryption/decryption of burst payloads. The LLC will append the burst header to a burst payload.

The MAC layer provides peer LLC layers with the ability to exchange MAC service data units (MSDUs=LPDUs=burst). On the transmission path, the MAC layer performs the grouping of bursts=MSDU into an MPDU. The MAC layer also provides the layer management with the ability to generate specific MPDUs=frames for performing background tasks: silence management, channel estimation, node status polling and node discovery. The MAC layer handles 7 types of frames (data, silence, channel estimation, polling, access, access reply and non-returnable data). All these frames start with a token announce delimiter. Except for the channel estimation frame, all the other frames are terminated with a token delimiter which content depends on the type of the frame.

The PHY layer performs the OFDM modulation and the digital signal processing (DSP) needed to transmit the MPDU over the PLC channel. It also adds the FEC redundancy.

3 PHY

This section specifies the Physical Layer Entity for an Orthogonal Frequency Division Multiplexing (OFDM) system. OFDM has been chosen as the modulation technique because of its inherent adaptability in the presence of frequency selective channels, its resilience to jammer signals, its robustness to impulsive noise and its capacity of achieving high spectral efficiencies.

3.1 Overview

The diagram in Figure 3 shows an example PHY layer of a transmitter.

On the transmitter side, the PHY layer receives its inputs from the Media Access Control Layer. Two separate bit streams are shown because of the different encoding for data and control information. The delimiters shall be interleaved by the Control Interleaving block and mapped by the HURTO Mapping block; while the data stream shall be mapped using the HURTO Mapping block or the Adaptive Mapping block, depending on the reliability level required for the transmission. The outputs of both Mapping blocks lead into the Scrambler, which is followed by 4D-TCM (Four Dimensional Trellis Coded Modulation) encoding. The next step is the OFDM Modulation, which is composed of the Subcarrier Modulator, the IDFT (Inverse Discrete Fourier Transform), the Cyclic Prefix generator and the symbol windowing operation structure. The resulting digital signal is converted to an analog signal by means of a DAC and IQ modulated, before passing to the amplifying and filtering stages.

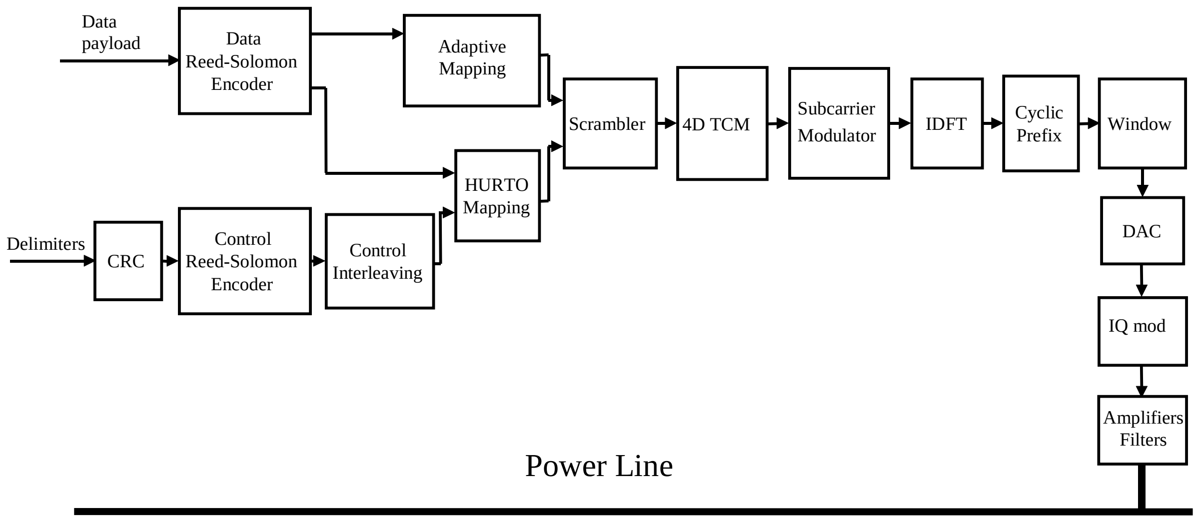

Figure 3 Transmission PHY layer

3.2 PHY layer frame

Figure 4 shows the symbols that are transmitted in a PPDU.

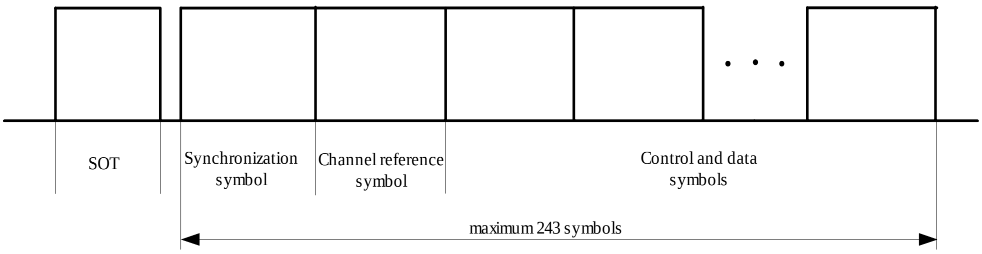

Figure 4 PPDU

The maximum duration of a PPDU is 243 symbols (counting from the synchronization symbol to the last symbol of the frame). A receiver shall be able to handle a maximum duration PPDU.

The gap between the last non-zero sample of the SOT and the first non-zero sample of the synchronization symbol shall be 2.75 ± 0.25 µs.

The time between two consecutive PPDUs is fixed and known by all modems as the TX/RX_SWITCH_TIME (see 3.11)

3.3 Forward error correction

3.3.1 Delimiters

There are several types of delimiters that will be explained in 4.4 and 5.2. Delimiters are mapped to one OFDM symbol, so they will always have the same size.

Three main calculations will be made over a delimiter as is shown in Figure 5.

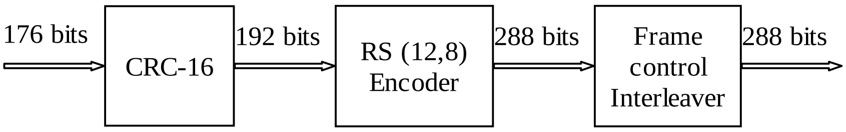

Figure 5 Delimiter encoding

3.3.1.1 CRC-CCITT

A CRC-CCITT is applied to the incoming 176 bits length delimiter applying the polynomial generator:

$$g(x) = x^{16} + x^{12} + x^{5} + 1$$

Equation 1

The CRC check called CRC(x), can be obtained as the remainder of the division between $I(x)*x^{16}$ and g(x), where I(x) is a binary coefficients polynomial given by:

$$I(x) = I_{175}x^{175} + I_{174}x^{174} + ... + I_2x^2 + I_1x + I_0$$

Equation 2

Therefore the CRC is expressed as:

$$CRC(x) = CRC_{15}x^{15} + CRC_{14}x^{14} + ... + CRC_1x + CRC_0$$

Equation 3

The resulting frame will be:

$$T(x) = I(x)x^{16} + CRC(x) = I_{175}x^{191} + I_{174}x^{190} + ... + I_2x^{18} + I_1x^{17} + I_0x^{16} + CRC_{15}x^{15} + ... + CRC_0$$

Equation 4

A total of sixteen bits resulting from the CRC algorithm are then appended to the delimiter.

The transformation is shown in the following figure:

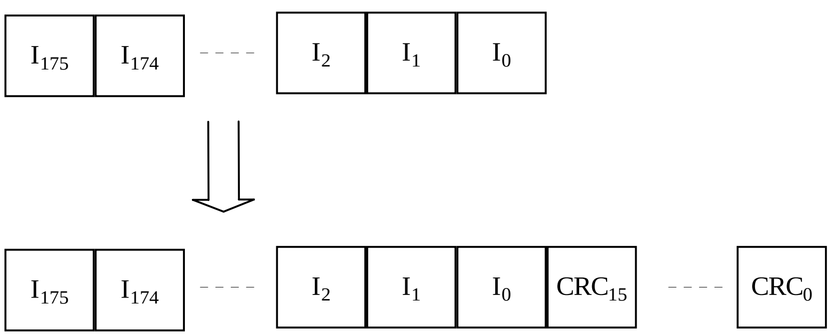

Figure 6 Delimiter CRC

3.3.1.2 Reed-Solomon Delimiter Coding

Delimiters are composed by six RS codewords from (n=12,k=8,t=2) RS code.

Four (n-k=4) parity symbols p_3, p_2, p_1, p_0 shall be appended to k=8 message symbols m_7, m_6, … , m_0 to form a Reed-Solomon codeword m_7, m_6, … , m_0, p_3, p_2, … , p_0, where symbol m_7 is the first four bits symbol in time out of the Reed-Solomon encoder. Each RS symbol is composed of four bits.

The parity symbols shall be computed from the message symbols using the Equation 5:

$$RS(x) = M(x)x^4 \bmod g(x)$$

Equation 5

Where the message polynomial is:

$$M(x) = m_7x^7 + m_6x^6 + ... + m_1x + m_0$$

Equation 6

The parity symbols are expressed as the following polynomial;

$$RS(x) = p_3x^3 + p_2x^2 + p_1x + p_0$$

Equation 7

The field generator binary polynomial that generates each RS symbol is given by:

$$f(x) = x^4 + x + 1$$

Equation 8

And the code generator polynomial is given by:

$$g(x) = (x-\alpha^1)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4)$$

Equation 9

The RS binary symbol representation for delimiters of α^0 is 0b0001, where the left most bit of this RS symbol is the most significant bit (MSB). This way, the binary representation of each of the redundancy symbols generated on the Galois field GF(2^4) denoted as p_3, p_2, p_1, p_0 will be represented as RSX_{15}, RSX_{14}, RSX_{13}, and RSX_{12} for the symbol p_3, RSX_{11}, RSX_{10}, RSX_9, and RSX_8 for the symbol p_2, RSX_7, RSX_6, RSX_5, and RSX_4 for the symbol p_1 and RSX_3, RSX_2, RSX_1, and RSX_0 for the symbol p_0, where *X* denotes the RS codeword number.

The following figure shows the final binary level representation of the delimiter:

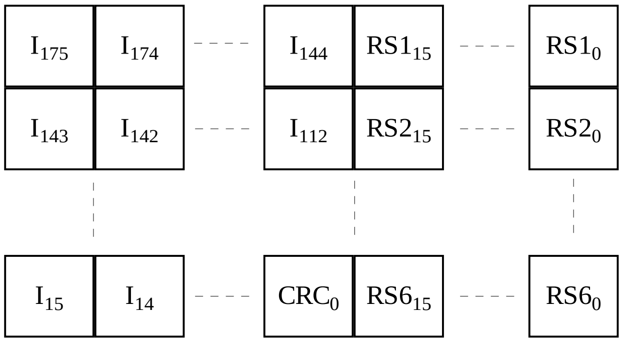

Figure 7 Reed Solomon bit format

The first bit to be transmitted is the leftmost of the first codeword (I_{175}), followed by the rest of the bits of the RS codeword. Then, the second codeword, and so on. This is illustrated in

$$I_{175}I_{174}...RS1_0\,I_{143}...RS2_0\,I_{111}.......RS5_0\,I_{15}...RS6_0$$

Figure 8 Reed Solomon bit order

3.3.1.3 Delimiter Interleaving

After RS encoding, the six resulting RS codewords shall be interleaved with interleaving depth of six RS codewords and 4 bits granularity, in such a way that the first four bits of each RS codeword shall be sent first to the interleaver, then, the second 4 bits of each of the six RS codewords and so on. We can group the bits in nibbles for clarity:

$$\begin{gathered}
N_{71} = (I_{175}, I_{174}, I_{173}, I_{172}) \\
\vdots \\
N_{64} = (I_{147}, I_{146}, I_{145}, I_{144}) \\
N_{63} = (RS1_{15}, RS1_{14}, RS1_{31}, RS1_{12}) \\
\vdots \\
N_{60} = (RS1_3, RS1_2, RS1_1, RS1_0) \\
\vdots \\
N_1 = (RS6_7, RS6_6, RS6_5, RS6_4) \\
N_0 = (RS6_3, RS6_2, RS6_1, RS6_0)
\end{gathered}$$

Equation 10

Figure 9 illustrates this interleaving

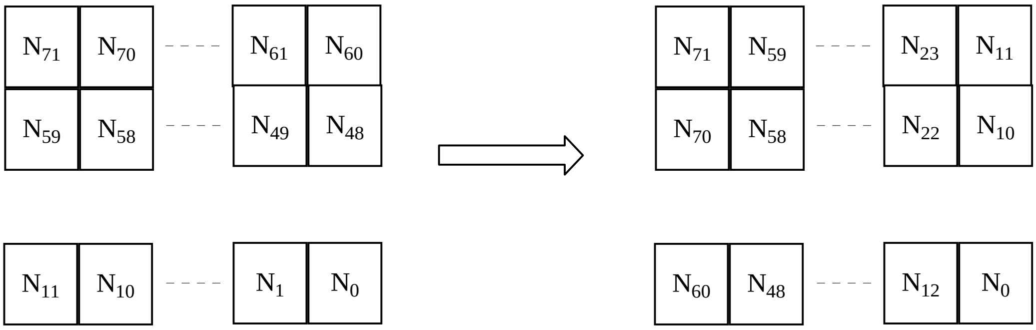

Figure 9 Frame control interleaving

With reference to the right half of Figure 9 the transmission order is row-wise, starting with the first row and from left to right, then the second row, and so on. In other words, the MSB of N71 will be the first to be transmitted, after N71, the next will be N59, and N0 will be last.

3.3.2 Data payload

The data payload, delimited by a burst header, shall be encoded by a variable rate 8-bit symbol Reed Solomon encoder. The RS data mode is set from the fields “FEC redundancy” and “FEC payload” of the burst header. The “FEC redundancy” value will determine which of the four RS data modes is used. The generator polynomial and redundancy of each of the available modes is given in Table 1:

RS data mode	g(x)	RS data code
0	$g(x)=(x-\alpha)(x-\alpha^2)(x-\alpha^3)...(x-\alpha^8)$	(k_0+8,k_0,t=4)
1	$g(x)=(x-\alpha)(x-\alpha^2)(x-\alpha^3)...(x-\alpha^{12})$	(k_1+12, k_1,t=6)
2	$g(x)=(x-\alpha)(x-\alpha^2)(x-\alpha^3)...(x-\alpha^{16})$	(k_2+16, k_2,t=8)
3	$g(x)=(x-\alpha)(x-\alpha^2)(x-\alpha^3)...(x-\alpha^{20})$	(k_3+20, k_3,t=10)

Table 1 RS data polynomials

The “FEC payload” field in each burst header contains the number of 32-bit words that will be coded per RS codeword, and therefore determines the amount of payload bytes per codeword (k_i).

Maximum and minimum value for this field is given depending on the mode in Table 2:

RS data mode	Redundancy words	Max payload in words	Min payload in words	
			HURTO	Adaptive
0	2	61	2	34
1	3	60	3	33
2	4	59	4	32
3	5	58	5	31

Table 2 RS data payload

Each data payload shall be coded following a different RS coding scheme that will be configured in each Burst header. The selected coding scheme is a shortened Reed-Solomon code that can be processed as a systematic RS(255, 255-2*t_i,t_i) by adding 255-(k_i+2t_i) bytes set to zero before the information bits. After the RS coding procedure these null bytes shall be discarded, leading to RS codewords of n = ki+2t bytes

When the data payload length is not an integer number of Reed-Solomon codewords, the last Reed Solomon codeword will have a special coding given by the formula:

$$k_{lastcw} = \left[(4 * APL) \bmod (k_i + 2 * t) \right] - 2t$$

Equation 11

Where APL is the data payload length in number of 32-bit words including RS coding parity bits.

For the last codeword, no minimum value in k is applied.

The incoming data payload shall be grouped in L groups of k_i bytes, where k_i has been taken from the “FEC payload” field in burst header, and a last group of k_{lastcw} bytes, calculated from Equation 11, as can be seen in the following figure:

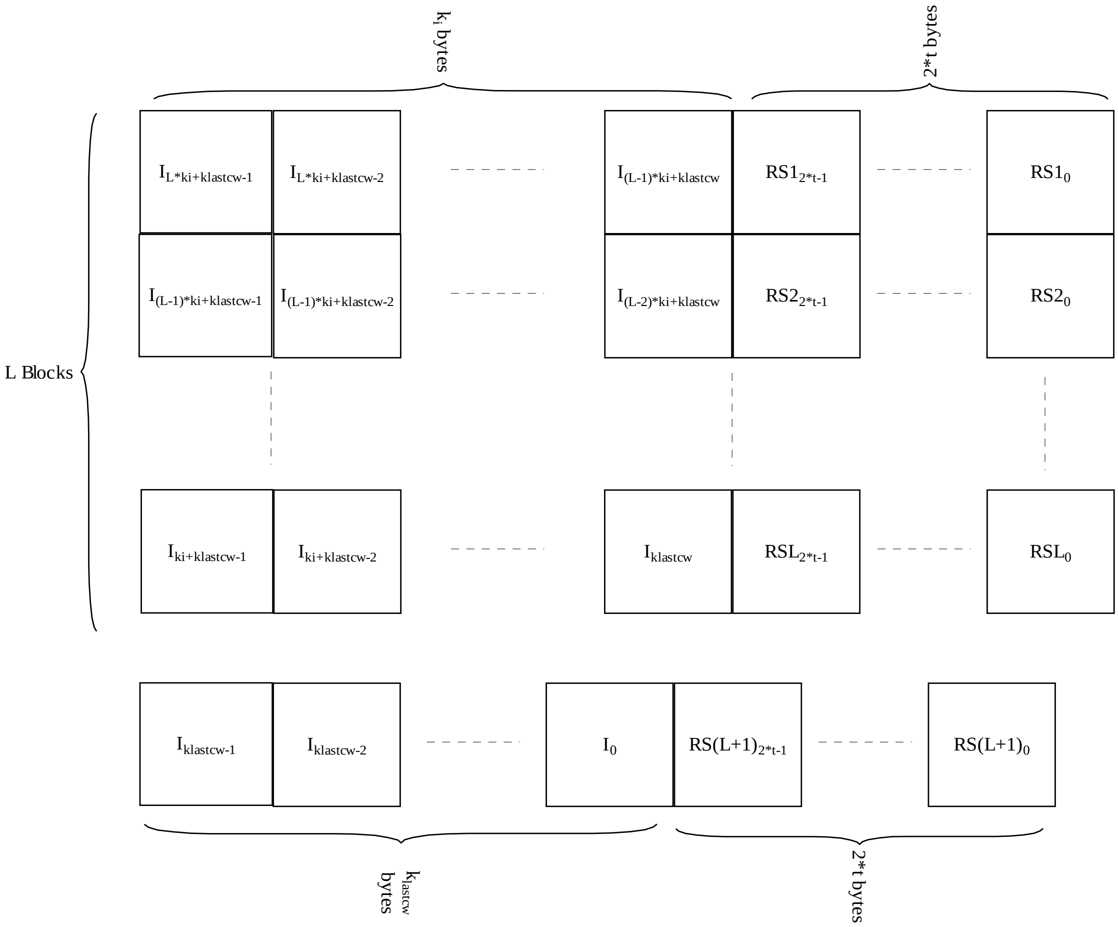

Figure 10 Reed Solomon byte grouping

L is given by the formula:

$$L = \left\lceil \frac{4 * APL}{(k_i + 2 * t)} \right\rceil$$

Equation 12

For each Reed-Solomon codeword, as in delimiter coding, (n-k) parity symbols p_{n-k-1}, p_{n-k-2}, … , p_0 shall be appended to k message symbols m_{k-1}, m_{k-2}, … , m_0 to form a Reed-Solomon codeword m_{k-1}, m_{k-2}, … , m_0, p_{n-k-1}, p_{n-k-2}, … , p_0, where symbol m_{k-1} is the first in time out of the Reed-Solomon encoder and the first in time of the uncoded data payload. Each of the symbols belongs to the Galois Field $GF(2^8)$, and it is then represented in its binary form with eight bits. On the other hand, n and k variables depend on the selected RS data mode, taken from Table 1 and Table 2. The parity symbols shall be computed from the message symbols using the equation:

$$P(x) = M(x)x^{n-k} \bmod g(x)$$

Equation 13

Where

$$M(x) = m_{k-1}x^{k-1} + m_{k-2}x^{k-2} + ... + m_1 x + m_0$$

Equation 14

Is the message polynomial,

$$P(x) = p_{n-k-1}x^{n-k-1} + p_{n-k-2}x^{n-k-2} + ... + p_1 x + p_0$$

Equation 15

Is the parity polynomial and g(x) is the code generator polynomial of the Reed-Solomon code, given by Table 1

The field generator polynomial associated with the Reed-Solomon code is given by:

$$f(x) = x^8 + x^4 + x^3 + x^2 + 1$$

Equation 16

The binary representation of α^0 is 0b00000001, where the left most bit of this RS symbol is the most significant bit (MSB).

3.4 Mapping modes

Once obtained a delimiter or a data payload, one from two different mapping procedures shall be used. The mapping mode is indicated with the field HURTO in the burst header (see section 5.2). HURTO mapping is fixed. Adaptive mapping will be done following a table (called bit-loading table) with information on how many coded bits will be allocated in each of the 1536 subcarriers in a symbol

3.4.1 HURTO mapping

HURTO mode shall be used in all delimiters and in the data payload if it is indicated in the HURTO field in the burst header.

First, groups of 288 consecutive bits are made. Each of these groups is replicated eight times, in order to obtain groups of 2304 bits. The bit allocation table will be set to 2 bits per subcarrier for all the 1536 subcarriers of an OFDM symbol, and each resulting 2304 bits groups shall complete an entire OFDM symbol. 4D-TCM encoding will be done, with a reset in convolutional encoder each 288 incoming bits.

In the data payload case with the HURTO bit set in the burst header, the last group will be completed with zeroes, if necessary.

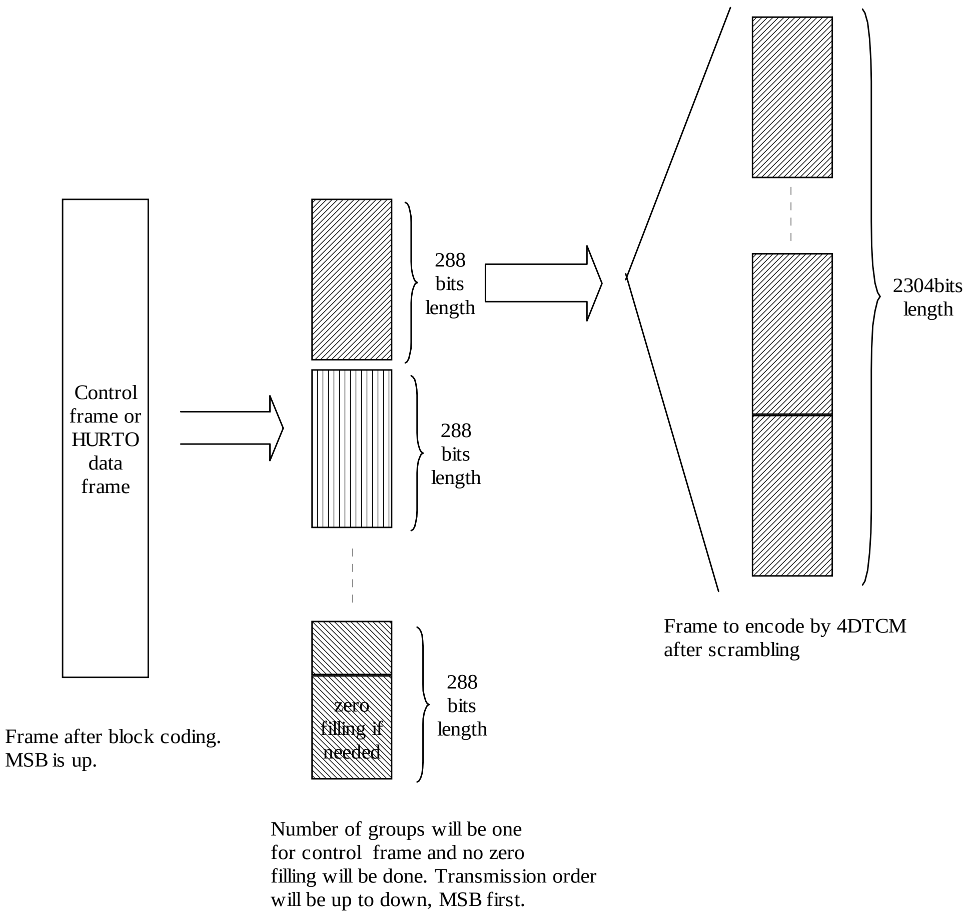

Figure 11 HURTO mapping

3.4.2 Adaptive mapping

When the HURTO bit is unset in the burst header, the data payload will be mapped following the bit allocation table, which sets how many coded (after 4D-TCM) bits will be allocated in each subcarrier.

When adaptive mapping is used, there must be at least 24 subcarriers with 2 or more information bits.

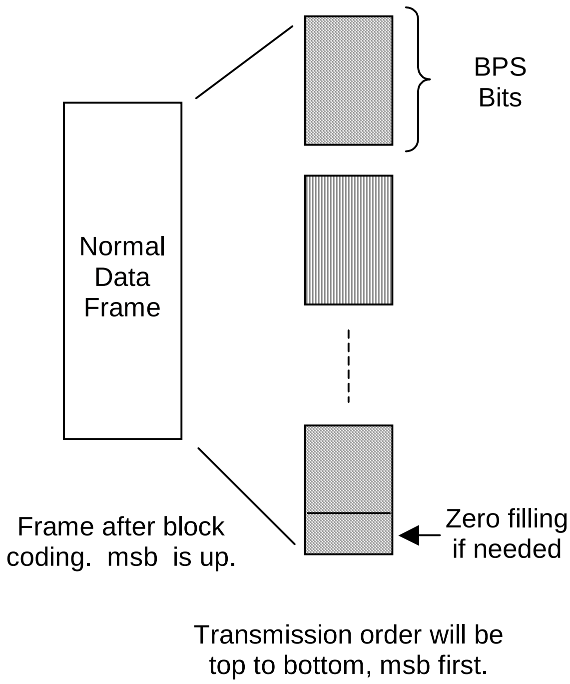

Figure 12 Adaptive mapping

BPS in the figure means the number of necessary data bits to fill an entire OFDM symbol with a certain bit-loading. A mathematical expression of this definition is shown in Equation 17. The last symbol shall be padded with zeros before scrambling, if necessary, until an integer number of symbols is used.

3.5 Trellis Coded Modulation (4D-TCM)

Once obtained a delimiter or data payload that will fill a integer number of OFDM symbols, a 16 state 4-dimensional truncated trellis code is used for error correction. This code will be applied over groups of 1536 subcarriers and reset each OFDM symbol, except in HURTO mode, where the 4D-TCM encoder will be reset each 192 subcarriers out, or, in other words, each 288 bits in.

3.5.1 Bit Extraction

Data bits from both delimiters and data payload shall be extracted according to a bit-loading table b_j, with j=0…1535, righter bits first, in groups of z bits. In the case of HURTO mapping $b_j = 2\ \forall j \in [0..1535]$. Due to the constellation expansion associated with coding and the 4-dimensional nature of the code, z_i will be calculated from a given pair (x,y) of two consecutive b_j, where b_j is the number of coded bits that subcarrier j will transmit.

Due to the heavy impairments of the powerline channel, or regional regulations, some subcarriers can be set not to carry any information. This case is denoted with a bit loading b_j = 0 for such subcarriers. When the two consecutive subcarriers x and y that form a symbol of the four dimensional code are set to zero, no bit extraction is performed, and the following pair of subcarriers is taken into account. On the other hand, when only one of the two subcarriers is set to zero, a special bit extraction is performed to obtain the word called t', including the insertion of two zeroes in position 0 and 2 of the final t' word to be introduced in the codec.

The extracted group will be noted as t' and calculated from the following table:

Condition	Binary word/comment
$x \geq 2 \quad y \geq 2$	$t' = (t_z \quad t_{z-1} \quad \ldots \quad t_4 \quad t_3 \quad t_2 \quad t_1)$ Where $z' = z = x + y - 1$
$x = 0 \quad y \geq 2$	$t' = (t_z \quad t_{z-1} \quad \ldots \quad t_2 \quad 0 \quad t_1 \quad 0)$ Where $z' = z + 2 = y + 1$
$x \geq 2 \quad y = 0$	$t' = (t_z \quad t_{z-1} \quad \ldots \quad t_2 \quad 0 \quad t_1 \quad 0)$ Where $z' = z + 2 = x + 1$
$x = 0 \quad y = 0$	Bit extraction not necessary, no message bits being sent

Table 3 Bit extraction

Since the maximum number of bits in a group will be 19, a 19 bits wide word called t´ will be formed from each two subcarriers, filling most significant bits with zeroes when $z' < 19$. This word will be scrambled with a pseudorandom word generator.

The bits per symbol (BPS) is defined by:

$$BPS = \sum_i z_i \quad \forall i \in Symbol$$

Equation 17

3.5.2 Scrambling

Each of the bits from the pseudorandom generator shall be generated with the same polynomial structure initialized with different seeds for each bit (where bit 0 is the least significant bit).

The pseudorandom sequence generator shall be the 10-bit PN generator with characteristic polynomial given by Equation 18.

$$f(x) = 1 + x^3 + x^{10}$$

Equation 18

At the beginning of each symbol, the registers of the PN generators shall be set with the corresponding seeds given by Table 4

PN	Seed
0	0x2bf
1	0x3f0
2	0x3ff
3	0x28d
4	0x2d7
5	0x017
6	0x1e4
7	0x3d8
8	0x337
9	0x1f1

10	0x2a3
11	0x3f3
12	0x000
13	0x11f
14	0x329
15	0x177
16	0x331
17	0x119
18	0x350

Table 4 Scrambling seed values

All the pseudorandom generators will advance once for each pair of extracted carriers. If no bit extraction is performed (x=y=0) the pseudorandom generators will not advance. The structure of each pseudorandom generator that generates a u word from the extracted t' word is shown in Figure 13

The symbol number value shall be 1 for the channel reference symbol, and shall be incremented by 1 for every symbol there after.

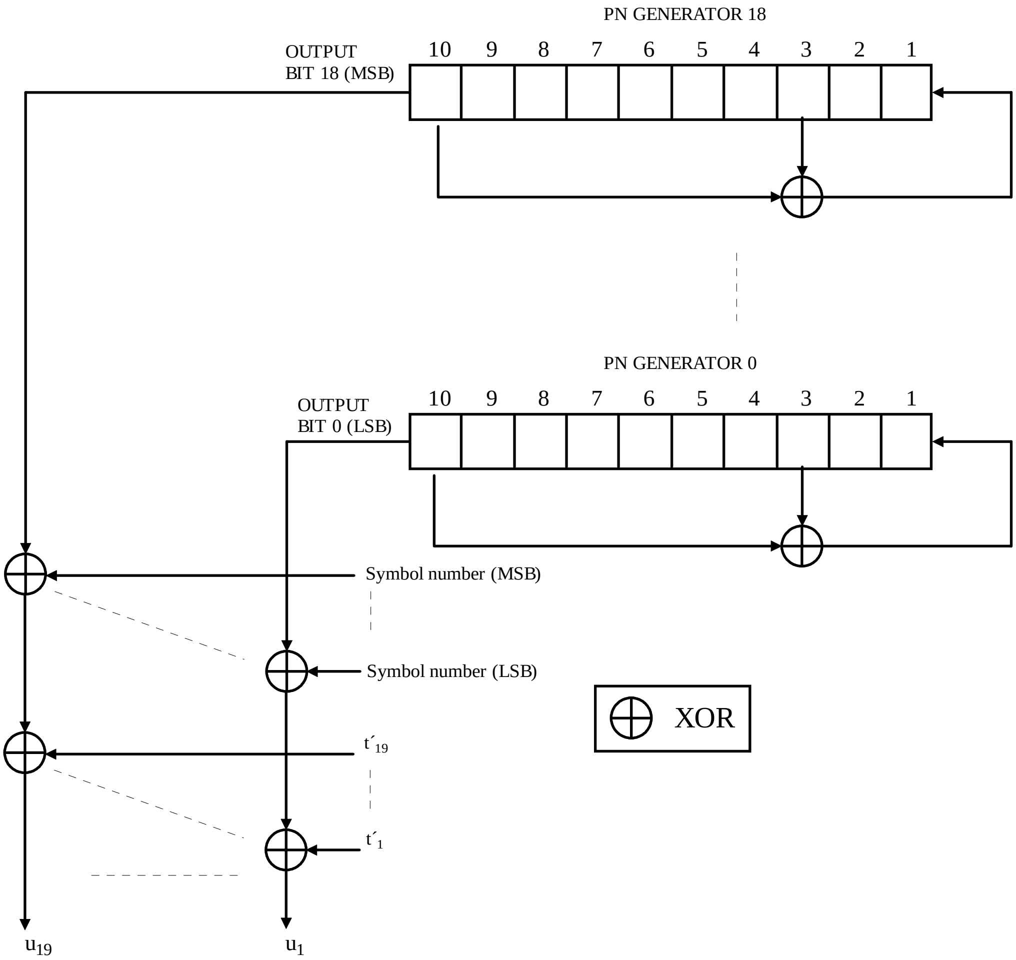

Figure 13 Scrambling

3.5.3 Bit conversion

The binary word u = ($u_{z'}$, $u_{z'-1}$,… u_1) determines two binary words v'=($v'_{z'-y}$,…,v'_0) and w'=(w'_{y-1},…,w'_0), which are used to look up two constellations points in the encoder constellation table. As a first step, v and w are obtained from u, as shown in Figure 14. Finally, v' and w' are obtained from v and w. For the usual case of x≥2 and y≥2, z´=z=x+y-1, and v´=v and w´=w contain x and y bits respectively. For the special case of x=0 and y≥2, z´=z+2=y+1, v´=v=(v_1,v_0) and w´=w=(w_{y-1},…,w_0), while when y=0 and x≥2, z´=z+2=x+1, w´=v=(v_1,v_0) and v´=w=(w_{x-1},…,w_0).

The bits (u_3, u_2, u_1) determine (v_1,v_0) and (w_1,w_0) according to following figure:

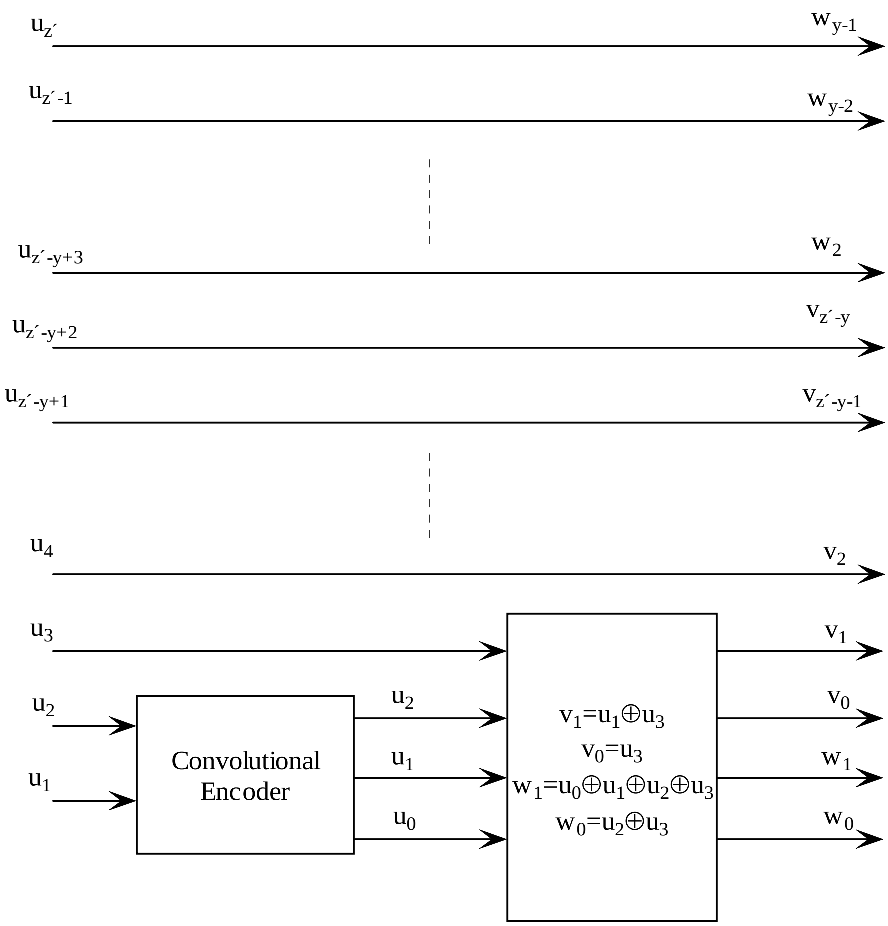

Figure 14 Conversion of *u* to *v* and *w*

The convolutional encoder shown in Figure 14 is a systematic encoder as shown in Figure 15. At the beginning of the encoding of each subcarrier group (1536 or 192 when HURTO mode is active), the convolutional encoder state is initialized to (0,0,0,0). No special operation to terminate the code is performed.

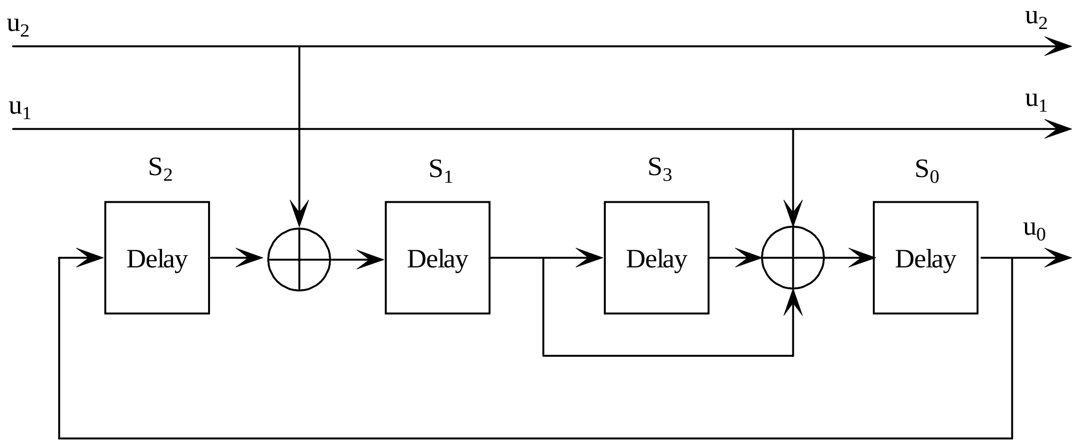

Figure 15 Convolutional encoder

The remaining bits of v and w are obtained from the less significant and more significant parts of ($u_{z'}$, $u_{z'-1}$,…, u_4), respectively. When x≥2 and y≥2, v´=v=($u_{z'-y+2}$, $u_{z'-y+1}$,..., u_4,v_1,v_0) and w´=w=($u_{z'}$, $u_{z'-1}$,…, $u_{z'-y+3}$,w_1,w_0). When x=0 or y=0 (but not both), v= (v_1, v_0), and it will be assigned to the zero loading subcarrier (v´ when only x=0 but not y and w´ when y=0 but not x). w will have y bits in the first case and x bits in the second, and will be assigned to the non zero loading subcarrier.

The binary word v´ is input first to the constellation encoder, and the binary word w´ last.

3.5.4 Constellation encoder

Each of the two resulting binary words after 4D-TCM process, called v' and w', represent an amplitude a_k and an increment to previous phase, called Δb_k that will perform a new constellation point to be transmitted in one subcarrier. The first word to be processed and transmitted will be v' and then w'. Since the mapping process to amplitude and phase increment will be the same for v' and w', we consider the following description based in a binary word called I that can be v' or w' indistinctly.

In the following table, mapping for each bit-loading and the pair (a_k, Δb_k) and its relationship with the binary word I is shown.

Bit-loading	a_k	Δb_k
2	0,0,0,0	0,0,0,0,I_1, $I_1 \oplus I_0$
3	0,0,0,I_1	0,0,0,0,I_2, I_0
4	0,0,0,I_1	0,0,0,I_2, $I_2 \oplus I_3$, I_0

5	0,0,I_4, I_1	0,0,0,I_2, $I_2 \oplus I_3$, I_0
6	0,0,I_5, I_1	0,0,I_2, $I_2 \oplus I_3$, $I_3 \oplus I_4$, I_0
7	0,I_6,$I_6 \oplus I_5$, I_1	0,0,I_2, $I_2 \oplus I_3$, $I_3 \oplus I_4$, I_0
8	0,I_7, $I_7 \oplus I_6$, I_1	0,I_2, $I_2 \oplus I_3$, $I_3 \oplus I_4$, $I_4 \oplus I_5$, I_0
9	I_8, $I_8 \oplus I_7$, $I_8 \oplus I_7 \oplus I_6$, I_1	0,I_2, $I_2 \oplus I_3$, $I_3 \oplus I_4$, $I_4 \oplus I_5$, I_0
10	I_9, $I_9 \oplus I_8$, $I_9 \oplus I_8 \oplus I_7$, I_1	I_2, $I_2 \oplus I_3$, $I_3 \oplus I_4$, $I_4 \oplus I_5$, $I_5 \oplus I_6$, I_0

Table 5 Constellation encoder

Where the binary word I will be first v´and then w´ till end all the 1536 subcarriers.

If no bit extraction is performed (x=y=0) the output of the constellation encoder shall be a_k= (0,0,0,0) and Δb_k =(0,0,0,0,0,0) for both subcarriers.

3.6 Subcarrier modulation

Each subcarrier of the Control or Data OFDM symbols shall be modulated using the ADPSK (Amplitude Differential Phase Shift Keying) modulation.

Equation 19 defines the ADPSK constellation of *M* phases and *N* rings (*MxN* ADPSK):

$$s_k = \lambda\left(A + a_k\right)e^{j\theta_k}$$

Equation 19

Where:

- k is the time index representing the k-th OFDM symbol. Index 0 corresponds to the channel reference symbol (see 3.8.1).
- s_k is the modulator output for a given subcarrier.
- λ is the factor to normalize the constellation to unit power:

$$\lambda = \frac{1}{\sqrt{A^2 + (N-1)\left(A + \frac{2N-1}{6}\right)}}$$

Equation 20

- a_k, $a_k \in \{0,1,\ldots,N-1\}$, represents the information coded in the amplitude, as supplied by the constellation encoder (see 3.5.4).

- θ_k stands for the absolute phase of the modulated signal obtained as follows:

$$\theta_k = \left(\theta_{k-1} + \left(2\pi/M\right)\Delta b_k\right) \bmod 2\pi$$

Equation 21

Equation 21 applies for k>0 only. Δb_k, $\Delta b_k \in \{0,1,\ldots,M-1\}$, represents the information coded in the phase increment, as supplied by the constellation encoder (see 3.5.4).

- *A* is a shaping parameter and represents the ring offset from the centre of the constellation.

Parameters *M*, *N* and *A* are determined by the number of bits per subcarrier indicated by the bit-loading table according to Table 6.

Bit-loading	Constellation points	*M*	*N*	*A*	Bits coded in the phase increment	Bits coded in the amplitude
2	4	4	1	1	2	0
3	8	4	2	0.8	2	1
4	16	8	2	1.7	3	1
5	32	8	4	1.7	3	2
6	64	16	4	2.5	4	2
7	128	16	8	2.6	4	3
8	256	32	8	5	5	3

9	512	32	16	4.6	5	4
10	1024	64	16	8.6	6	4

Table 6 Constellation parameters

The 4D-TCM provides the a_k and Δb_k values to the ADPSK Modulator according to the bit-loading table. If the bit-loading table indicates that a subcarrier does not carry any information (0 bits per subcarrier) this subcarrier shall be modulated using 2 bits.

3.7 OFDM modulation

This section describes how the input constellations are grouped to create OFDM symbols. Special symbols for synchronization and SOT are also described.

The signals will be described in a complex base-band signal notation. Different implementations may be possible, if the signal transmitted to the channel is the same.

3.7.1 Control and Data symbols

There shall be three symbol types that will define the signal bandwidth used by the system. The three types of symbols will be referred to as Type I (30 MHz signal bandwidth), Type II (20 MHz signal bandwidth) and Type III (10 MHz signal bandwidth). The symbols are defined based on three system clocks that will define the signal bandwidth. A node shall be operating in one of the three types at a given time. Specifically, symbol types are not mixed in a PPDU, and the transition from the use of a symbol type to a different one can take in the order of seconds.

Table 7 lists some timing parameters of the OFDM modulation.

Parameter	**Type I**	**Type II**	**Type III**	**Units**
System clock	40	$26.\widehat{6}$	$13.\widehat{3}$	MHz
N_{SD}: Number subcarriers	1536	1536	1536	Carriers
N_{IDFT}: IDFT interval	2048	2048	2048	Samples
T_{IDFT}: IDFT interval	51.2	76.8	153.6	µs
N_{CP}: Cyclic prefix	800	532	268	Samples
T_{CP}: Cyclic prefix	20	19.95	20.1	µs

N_{SYM}: Symbol interval	2848	2580	2316	Samples
T_{SYM}: Symbol interval	71.2	96.75	173.7	µs

Table 7 OFDM modulation parameters

The stream of complex numbers coming from the subcarrier modulator is divided into groups of N_{SD}=1536 to form OFDM symbols.

If a complex 2048-point IDFT is used, the 1536 subcarriers shall be mapped in the way shown in Figure 16.

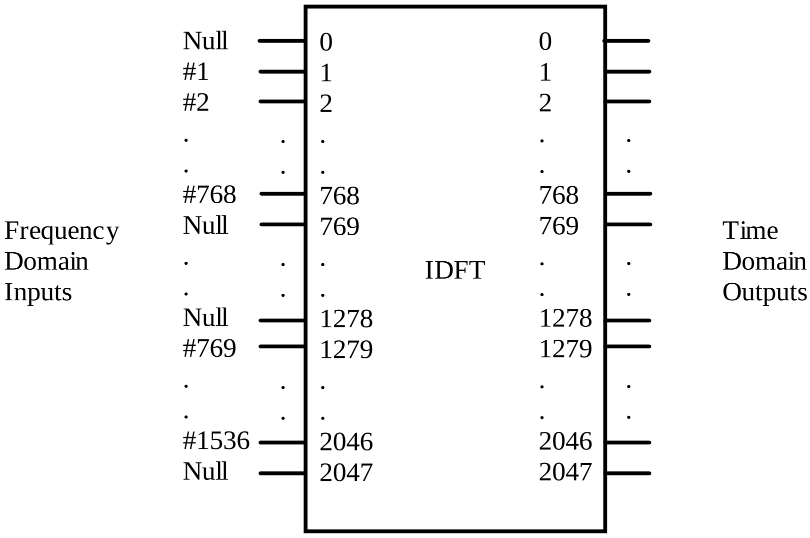

Figure 16 Subcarrier mapping

After the inverse Fourier transform, the symbol is cyclically extended by N_{CP} samples to create the "cyclic prefix".

3.7.2 Time domain windowing

The symbol may be multiplied by a windowing function to smooth transitions between the symbols. However, the binding requirement is the spectral mask as detailed in 3.13. Time domain windowing is just one way to achieve that goal. The implementer may use other methods to achieve the same goal such as time domain filtering. Therefore the transition shape and duration (N_W) are not specified here. In the particular case where N_W=0 the window degenerates into a rectangular pulse of value 1 and duration N_{SYM}. In the general case where N_W>0 the window extends over more than one symbol (N_{SYM}+N_W) and the symbols overlap as shown in Figure 17. The general expression for the windowing function is given in Equation 22.

$$w(n) = \begin{cases} f(n) & 0 \le n < N_W \\ 1 & N_w \le n < N_{SYM} \\ f(N_W + N_{SYM} - n - 1) & N_{SYM} \le n < N_W + N_{SYM} \\ 0 & elsewhere \end{cases}$$

Equation 22

And the windowing function fulfils the following condition:

$$\mathrm{f(n) + f(N_w - n - 1) = 1}$$

Equation 23

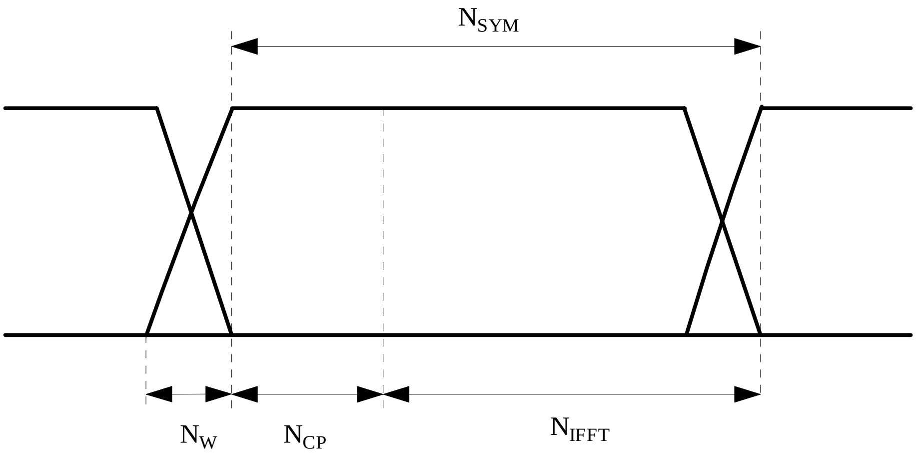

Figure 17 Windowing

The maximum duration of the window (N_w) depends on the symbol Type used:

- Type I: 128 samples
- Type II: 128 samples
- Type III: 64 samples

The OFDM symbol can be expressed in mathematical form as in Equation 24.

$$s_i(n) = k_{SYM}\, w_{SYM}(n) \left\{ \sum_{k=1}^{768} c(k,i) \exp\left(\frac{j2\pi k}{2048}(n - N_{CP} - N_W) \right) + \sum_{k=1279}^{2046} c(k-510,i) \exp\left(\frac{j2\pi k}{2048}(n - N_{CP} - N_W) \right) \right\}$$

Equation 24

Where

– i is the i-th OFDM symbol; i = 0, 1, …

- n is the sample index; $0 \leq n < N_{SYM}+N_W$
- k_{SYM} is a normalization factor, that shall be chosen in order to meet the PSD and modulation accuracy requirements.
- $w_{SYM}(n)$ is a windowing function
- c(k,i) is the $k+iN_{SD}$ complex value from the subcarrier modulation block

Several of these symbols are concatenated in a burst as expressed in Equation 25

$$s(n) = \sum_i s_i(n - iN_{SYM})$$

Equation 25

3.8 Reference signals

3.8.1 Channel reference symbol

The channel reference symbol comes before the data and control symbols. The channel reference symbol follows Equation 24 with

$$c(k,i) = \exp(j\phi_k)$$

Equation 26

where ϕ_k is a random phase picked from a uniform distributed random variable between $[0,2\pi[$. The channel reference symbol gives the initial reference for the subcarrier modulation (3.6)

3.8.2 Synchronization symbol

The synchronization symbol comes before the channel reference symbol. The synchronization symbol follows Equation 24 with

$$c(k,i) = \begin{cases} \sqrt{2}\exp(j\phi_k) & k \quad even \\ 0 & k \quad odd \end{cases}$$

Equation 27

where ϕ_k is a random phase picked from a uniform distributed random variable between $[0,2\pi[$.

3.8.3 Start Of Transmission (SOT)

A start of transmission signal shall always be sent before the synchronization symbol. The SOT is also used as response to the polling (4.3.4). In reception the SOT may be used for AGC. The description of the SOT generation is always based on a 40 MHz clock, independently of the symbol type used by the node. The number of active subcarriers in the SOT depends on the symbol type used.

The SOT is composed of OFDM symbols that may be generated with a complex 256-point IDFT as shown in Equation 28.

$$s_{D,i}(n) = k_D w_D(n)\left\{\sum_{k=1}^{N_D/2} c(k)\exp\left(\frac{j2\pi k}{256}n\right) + \sum_{k=255-N_D/2}^{255} c(k)\exp\left(\frac{j2\pi k}{256}n\right)\right\}$$

Equation 28

Where

- k_D is a normalization factor
- $w_D(n)$ is a windowing function, that follows Equation 22 and Equation 23. The duration of the windowing function is between 6.4 µs and 8 µs depending on the transition duration and is depicted in Figure 18.

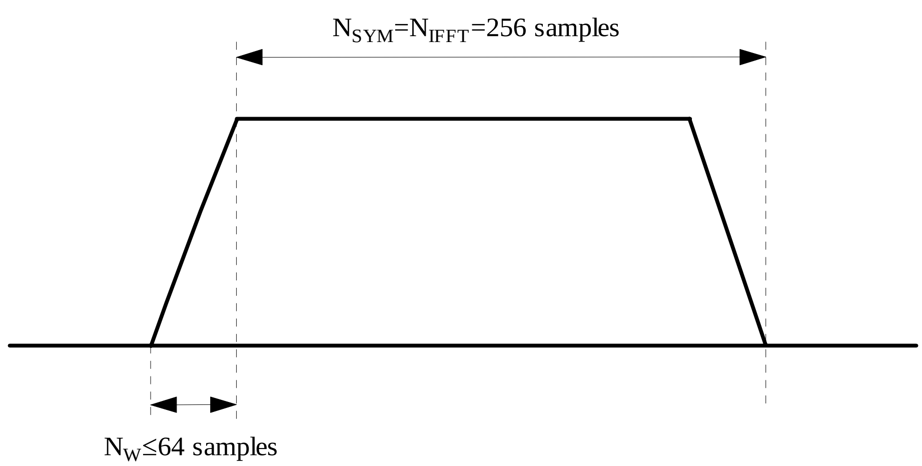

Figure 18 SOT Windowing

- N_D is 192 (Type I), 128 (Type II) or 64 (Type III), so that the frequencies occupied by the SOT are the same as for the data and control symbols.

- c(k) = exp(jϕ_k) where ϕ_k is a random phase picked from a uniform distributed random variable between [0,2π[

Six of these symbols are concatenated to form the SOT.

$$s_{SOT}(n) = \sum_{i=0}^{5} s_{D,i}(n - 256i)$$

Equation 29

Therefore the maximum duration of the SOT (if the maximum window size is used) will be 40 µs.

3.8.4 Channel estimation symbols

The channel estimation symbols are used to estimate the channel quality at the receiver side. After the channel estimation the receiver may send a new bit-loading table to the transmitter (see 9.1.2). These symbols shall be known by the transmitter and the receiver to allow a correct estimation under all conditions. Therefore, a pseudorandom sequence generator shall be used as the modulator input instead of the output from the 4D-TCM encoder.

The pseudorandom sequence generator shall be the 11-bit PN generator with characteristic polynomial given by Equation 30.

$$f(x) = 1 + x^2 + x^{11}$$

Equation 30

This symbol shall be modulated using 2 bits per subcarrier. Therefore, two 11-bit PN generators with different seeds shall be used. Figure 19 explains how the Δb_k value is formed. The SYMBOL NUMBER value is 1 for the channel reference symbol, and is incremented by 1 for every symbol there after.

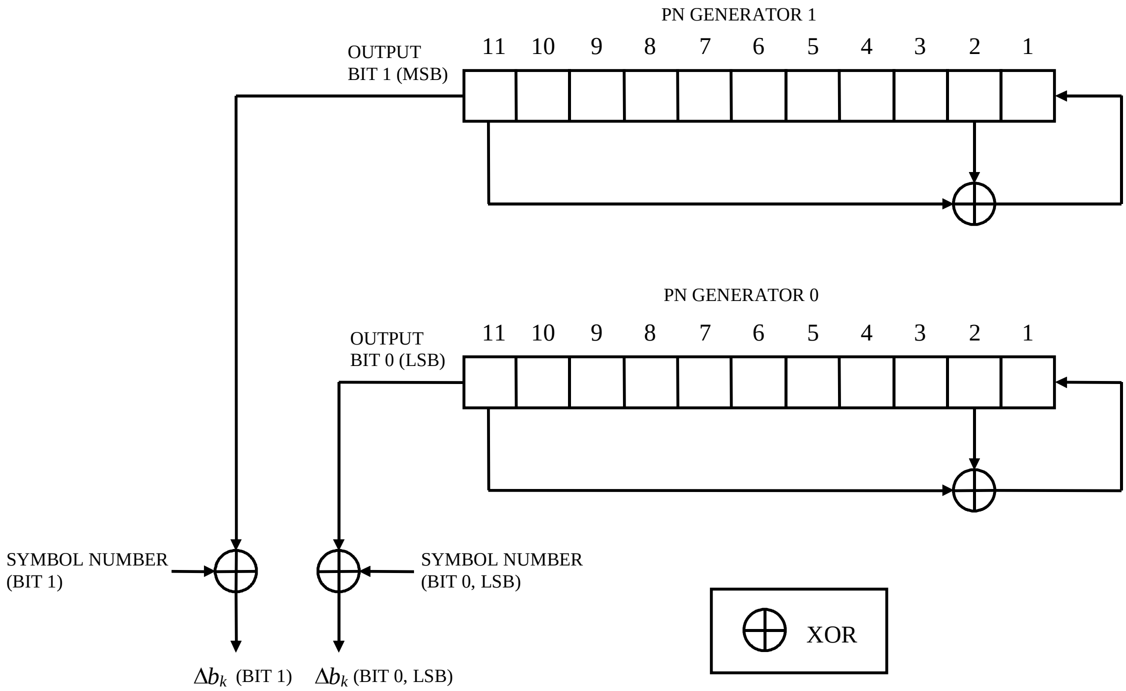

Figure 19 Channel estimation symbol generation

At the beginning of each Channel estimation symbol, the registers of the PN generators shall be set with the corresponding seed given by Table 8. So, the value to be modulated in the phase increment of the first valid subcarrier of the OFDM symbol will be determined by the seeds and the appropriate symbol number. The contents of the registers of the PN generators shall be updated before calculating the phase increment for each of the next subcarriers.

Since two bits of the symbol number are XORed with the output of the PN generators, four different symbols will be generated. These four symbols will be repeated four times to create a sixteen-symbol sequence.

	Seed
PN Generator 1	0x25E
PN Generator 0	0x3C1

Table 8 PN generator seeds

3.9 Carrier frequency

The complex baseband digital signal is converted to analog signal and IQ modulated. I(t) and Q(t) are the in-phase and quadrature analog signals corresponding to the real and imaginary components of the complex baseband signal respectively. The actual transmitted signal is related to them by the following relation:

$$r_{TX}(t) = I(t)\cos(2\pi f_c t) - Q(t)\sin(2\pi f_c t)$$

Equation 31

f_c denotes the carrier centre frequency, which must be a multiple of 156.25 KHz.

3.10 Communication modes

Transmission modes can be defined by specifying a carrier frequency (f_c), and symbol Type (I, II or III) and, optionally, a power mask. Any two nodes that use a common communication mode shall be able to communicate.

The carrier frequency shall be higher than the bandwidth of the baseband signal in negative frequencies in order to avoid ICI.

3.11 PHY parameter specification

Table 9 contains the values for the PHY parameters.

Parameter	**Value**	**Description**
TX-RX switch time	356 μs (Type I) 387 μs (Type II) 347.4 μs (Type III)	See 3.2 and 4.1

Table 9 PHY parameter specification

3.12 Clock synchronization

3.12.1 General

Clock frequency synchronization shall be performed and must be accurate enough to allow for maximum bit loading 10 bits on all subcarriers.

3.12.2 Clock synchronisation concept

With the exception of the HE of a PLC cell, all PLC nodes belonging to the same PLC cell must adapt their clock frequency until a unique common clock frequency is reached. This common clock frequency is the reference clock frequency dictated by the HE. All other nodes (Repeaters and CPEs) of the PLC cell will synchronize their clock frequency to this reference clock frequency.

Both transmitter and receiver of a PLC node shall use the same common clock frequency.

3.12.3 System reference clock error tolerance

Without noticeable loss of performance, PLC nodes shall be able to cope with a relative clock frequency error of up to twice the maximum absolute frequency error of the system reference clock. The absolute frequency error of the system reference clock in any PLC equipment shall not exceed ±100 ppm.

3.13 Transmitter Electrical Specification

The following specification establishes the minimum transmitter technical requirements for interoperability maintaining full performance. Unless otherwise stated, transmitter specifications assume a 50 Ω load between line and neutral terminals. All transmitter output voltages as well as spurious transmissions are specified as the voltage measured at the line terminal with respect to the neutral terminal.

3.13.1 Transmit PSD

The PSD in the emission point must be compliant with regulations in effect in the country where the system is used.

The ripple in the working band shall be less than ±1 dB.

Besides there will be a transmit spectrum mask that will depend on the regulation of each country or area. All nodes shall use a configurable power masking in order to be compliant with that mask achieving spectral notches with at least 30 dB deep.

The transmitter must provide means to modify the transmission power of each subcarrier with a granularity of at least 6 dB. Once the transmitter introduces the notches in the transmitted signal, other mechanisms like the ABLP (see 9.1.2) will automatically adapt accordingly.

3.13.2 Modulation error vector magnitude

Modulation errors may be caused by quantization in the digital processing and by linear and non-linear distortion in the analogue processing of a PLC transmitter.

The normalized mean modulation error vector magnitude is defined as the RMS value of the modulation error vector magnitude divided by the RMS value of the modulation vector.

- N denotes the number of subcarriers.
- $S(k)$ denotes the measured complex symbol of the k-th subcarrier.
- $S_0(k)$ denotes the ideal complex symbol of the k-th subcarrier.

The RMS error vector magnitude may be expressed as:

$$EVM = \sqrt{\frac{\frac{1}{N}\sum_{k=1}^{N}\left|S(k)-S_0(k)\right|^2}{\frac{1}{N}\sum_{k=1}^{N}\left|S_0(k)\right|^2}} \cdot 100\%$$

Equation 32

and can be measured with a multi-tone error vector magnitude meter.

The EVM shall not exceed 1%

3.13.3 Power Control

This specification allows for the optional implementation of a Power Control mechanism, using the BPS and Rx Att information supplied in the ABLP packet (9.1.2.3) and the Power Control bits in the Token Announce (4.4.1). The target of the power control is to transmit the lowest power that attains a given performance. Therefore a node may adjust its transmission power until the desired value of BPS is reached. The process is depicted in Figure 20.

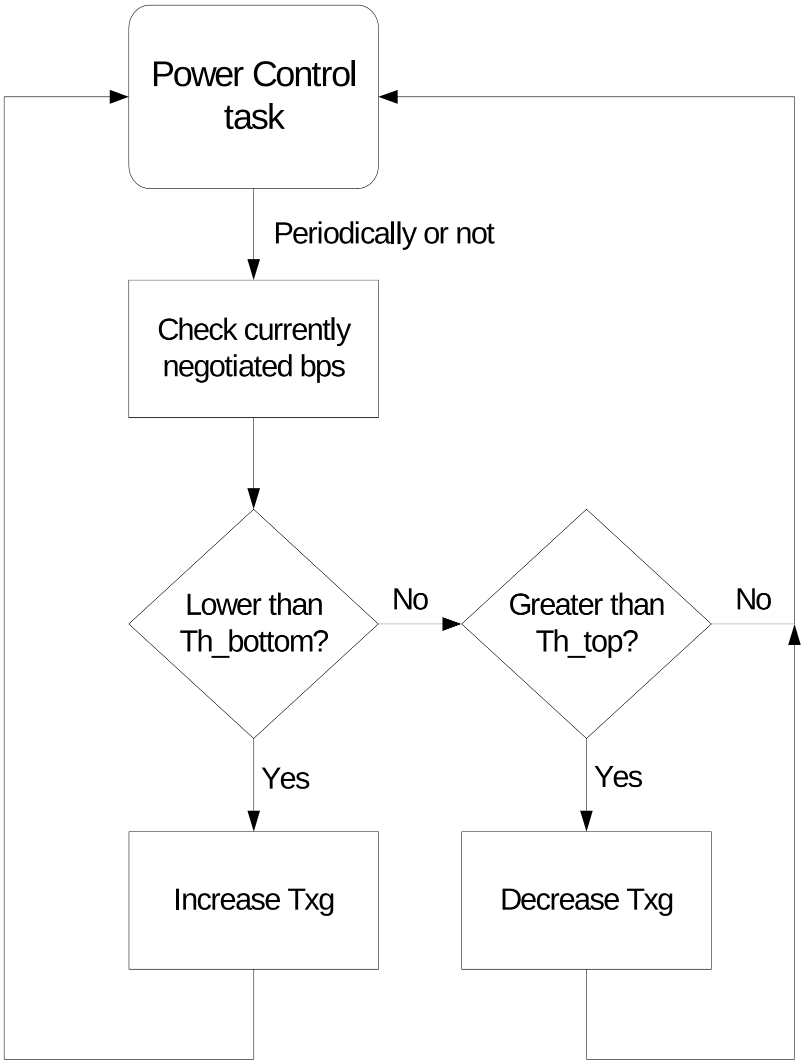

Figure 20 Power control algorithm

3.14 Receiver Electrical Specification

Unless otherwise stated, all signals are specified as the voltage measured at the line terminal with respect to the neutral terminal.

3.14.1 Receiver Input Impedance

When not transmitting, the system shall present a minimum impedance of 40 Ω in the working band, measured between line and neutral terminals.

3.14.2 Receiver Input Referred Noise

The system shall present an IRN level lower than –148 dBm/Hz when it sets the maximum reception gain.

3.14.3 Receiver Input Signal

The receiver shall maintain full performance with a signal input level of 8 V_{pp}.

The ripple introduced by the system in the working band shall be less than ±1 dB.

The sensitivity of the receiver is defined as follows:

For an input signal with flat spectrum and using Type I OFDM symbols, the minimum input level at which the system shall receive 2 Mbps of UDP traffic with 0% PLR is:

- 0.17 mV_{rms}
- -137 dBm/Hz referred to 50 Ω

4 MAC

4.1 Overview

The OPERA MAC is based on a TDMA/TDD MAC with hybrid resource sharing mechanisms built on top of an OFDM PHY layer. Data are encapsulated inside OFDM symbols. The control and data symbols transmitted consecutively by a single node constitute a transmission frame.

The channel access is done through the use of a special MAC packet called a token. Master nodes decide the type of frame that their slave nodes are going to transmit next, based on the type of token included in the current frame. Tokens have several intended uses, depending on the type of token, and the actions to be undertaken upon its reception depend on the type of token.

The master node manages the channel access, and decides which of its slaves will be allowed to transmit data, in what order, and for how long. These decisions will be published by the use of the data token. The data token has a specific destination node, which, upon its reception, must enter transmission mode. This destination node might then return the token to its master, or transmit it to its own slaves, in the case of a Time Division Repeater.

In a simple topology as depicted in Figure 21, the transmission scheme could be as depicted in Figure 22.

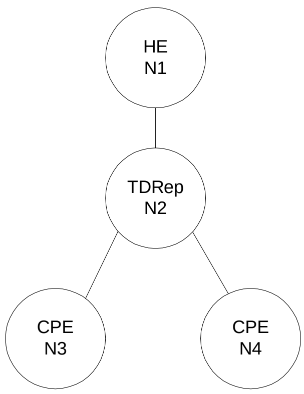

Figure 21 Simple Access Topology example

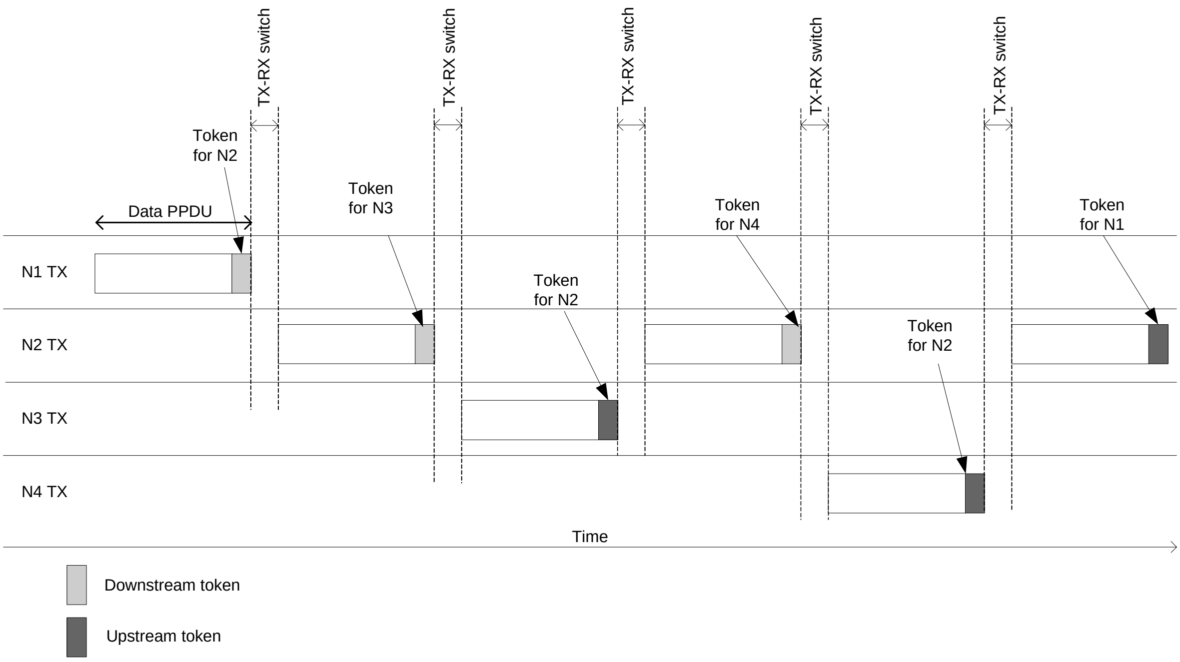

Figure 22 Transmission sequence

The diagram in Figure 22 depicts a typical transmission sequence. When N2 receives the token from its master, it is this node that must decide what nodes will follow: in this case, it transmits it to each of its slaves, and finally returns it to its master. Any other kind of sequence would be possible, and it is the master, at each level, who decides how the token, and thus the slots of time, will be divided amongst the rest of the nodes.

The maximum transmission time, known as validity, is variable (for minimum and maximum values allowed, see 4.6) and is decided by the master node for each frame. The actual transmission time for each frame might be smaller than the validity, if there is not enough data to fill it.

4.2 Slave states management

Slave states are specific states managed both on the master side and the slave side. They correspond to specific actions described here below.

4.2.1 Master side

From a master standpoint, a slave can be in three different states: Active, Idle, and Unregistered. These states are related to the following master-side actions (see Figure 23):

- Active: A master shall regularly transmit a data token to Active slaves. If a slave returns a data token without using any of the time resources originally granted, the master may list this slave as Idle.

- Idle: No data token is transmitted to an Idle slave. A master shall regularly transmit two types of Polling Tokens to an Idle slave.

    - Active Polling Tokens shall be regularly transmitted by the master to manage transitions from Idle to Active. A slave which replies to an Active Polling Token shall be listed as Active by its master. A master is constrained to transmit an Active Polling Token with a max interval equal to *MAX_ACTIVE_POLL_INTERVAL*.

    - Alive Polling Tokens shall be regularly transmitted by the master to manage transitions from Idle to Unregistered. A master is constrained to transmit an Alive Polling Token with a max interval equal to *MAX_ALIVE_POLL_INTERVAL*. A slave which does not reply to a number of Alive polling tokens= *MAX_ALIVE_TOKENS* shall be listed as Unregistered.

  Note: If a master has pending downstream data for an Idle slave, the master may automatically switch the status of this slave into Active.

- Unregistered: An Unregistered slave is not explicitly managed by a master. However, the master is constrained to regularly broadcast access frames to discover/recover Unregistered slaves. The max transmission interval of these access frames is *MAX_ACCESS_INTERVAL*.

4.2.2 Slave side

From a slave standpoint, a slave can be in two different states: Registered or Unregistered (see Fig 2)

- Unregistered: This is the initial state of a slave. An unregistered slave is only authorized to answer access frames as required by the access protocol. When the access protocol process is completed and successful, the slave enters the Registered state.

- Registered: The following specific requirements apply to this state:

    - A Registered slave shall never reply to access frames from its current master. It might reply to access frames from another master if it wishes to change its current master.

    - A Registered slave shall not reply to an Active polling token if it has no need for network resources.

    - A Registered slave that wishes to continue in that state shall always reply to an Alive polling token.

- A Registered slave which has not received any Alive polling token or data token for more than *MAX_ALIVE_TOKENS*MAX_ALIVE POLL_INTERVAL* shall enter the *Unregistered* state. The counting of this time interval shall be initialized at the end of the last polling slot in which the slave has asserted a SOT or at the reception of the last Data Token received by the Registered slave, whichever is later.

4.2.3 Transition state diagrams

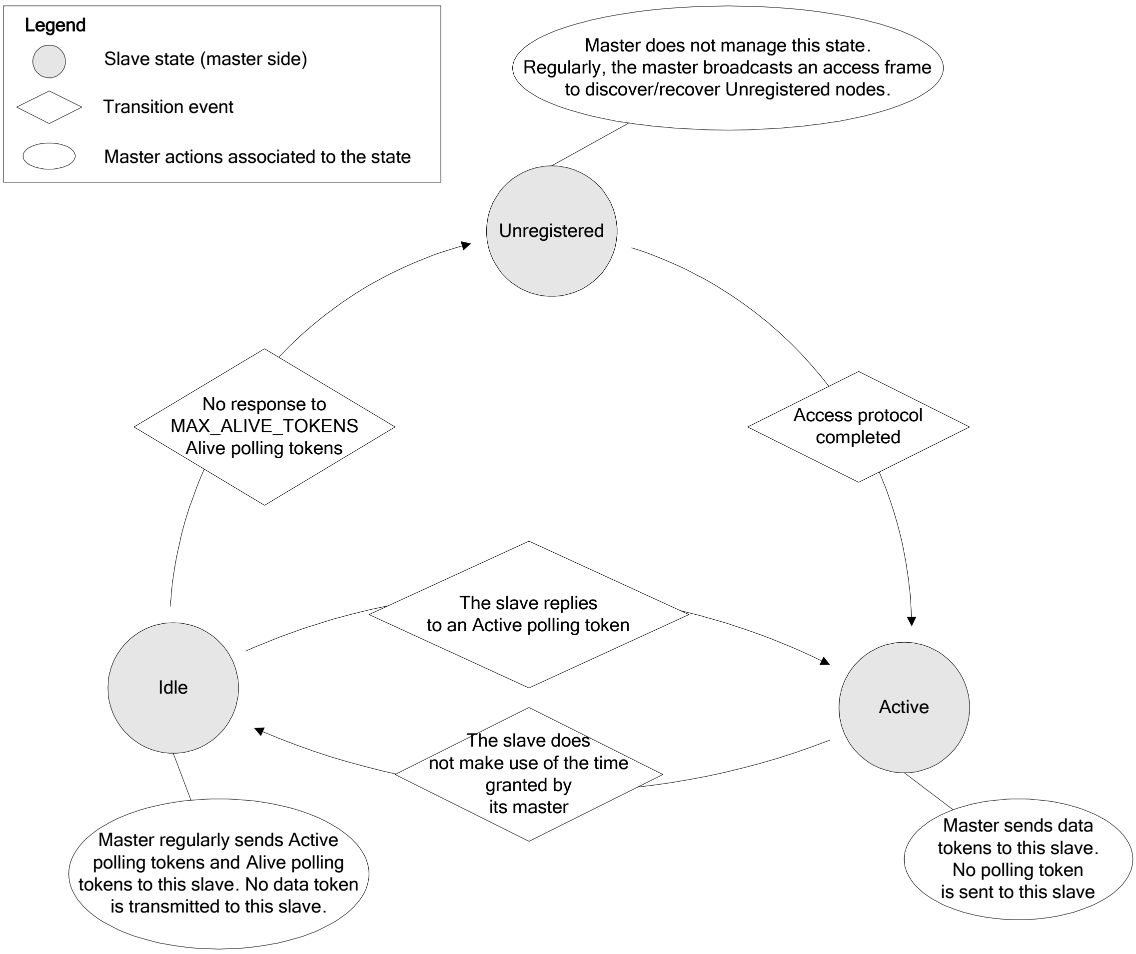

Figure 23 Slave states – Master side

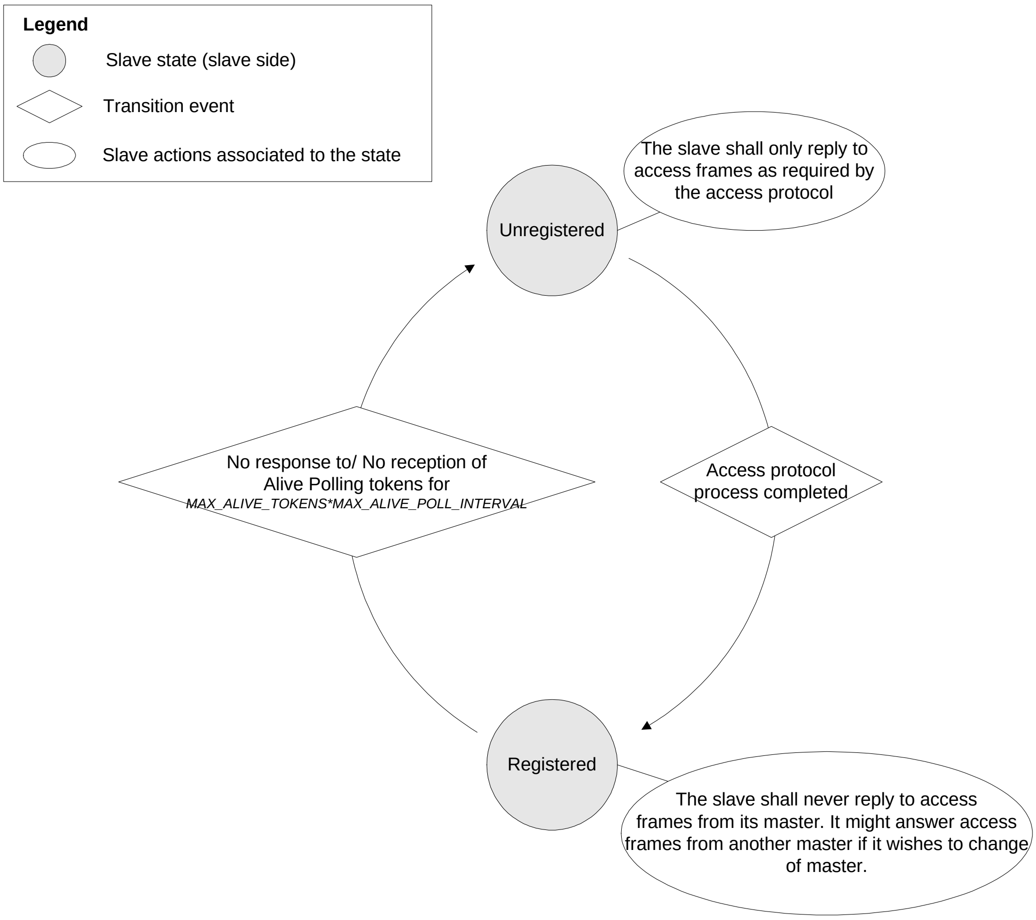

Figure 24 Slave states – Slave side

4.3 Frame formats

A frame corresponds to an MPDU passed by the MAC layer to the PHY layer. There are two types of MPDUs: regular MPDUs and Channel Estimation MPDUs.

Regular MPDUs can encapsulate bursts provided by the LLC layer.

Channel Estimation MPDUs are directly triggered by the MAC layer; they carry a specific signal sequence generated by the PHY layer: Channel Estimation MPDUs do not encapsulate any burst provided by the LLC layer.

Regular MPDU:

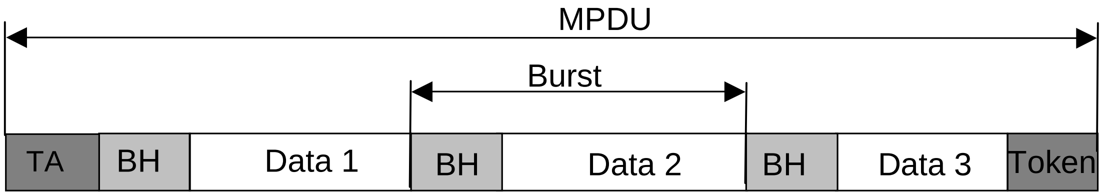

Figure 25 Regular MPDU format

The structure of a regular MPDU is depicted in Figure 25. The MPDU is of variable size and it is composed by:

- A token announce (TA) delimiter: The token announce defines, amongst other things, which is the current transmitter and what is the maximum transmission time of the current frame (Token Announce Validity).

- A sequence of bursts, each burst being composed of a Burst Header delimiter followed by data payload. The bursts included in a transmission frame can have different destination nodes. Burst addressed to a particular node shall be transmitted in order according to their Burst Id (see 5.2). Bursts to different destinations can be transmitted in any order. The destination port of the burst is included in the Burst Header, as well as information such as transfer control, encryption, used FEC, etc. The number of bursts in a regular MPDU ranges from zero to the maximum allowed by the assigned validity (see 4.4.1).

- A token delimiter.

The type of a regular MPDU is defined by the type of token terminating in the frame. The six types of regular MPDUs are:

1. Data frame
2. Silence frame
3. Polling frame
4. Access frame
5. Access Reply frame
6. Non-returnable Data frame

Channel Estimation MPDU

All MPDUs follow the sequence described above except for the Channel Estimation Frame: this frame does not contain neither bursts nor token. It starts with a Token Announce delimiter followed by a sequence of channel estimation symbols.

4.3.1 Data Frame

When a data frame is transmitted from a master to one of its slaves (the direction of the token is downstream), the data token at the end of the data frame will indicate for how long the slave is allowed to transmit. This length of time is known as the token validity, and is expressed as a number of symbols. The maximum length of the data frame is limited by the validity.

Upon transmitting a data token, the node will enter reception mode.

When a CPE receives a data token, it may transmit a data frame for as long as the received data token validity indicated. At the end of the frame, a data token is transmitted indicating the end of the transmission. At this point, the master node will take over the channel again.

When a Time Division Repeater (TDR) receives a data token from its master node, it schedules the transmissions between itself and the master node and also distributes new tokens among its slaves, for as long as the received data token validity indicated.

The validity does not need to be fully used by the receiving node, and it will not be wasted: the validity only sets the maximum length of time between the transmission of a token downstream and the return of this token, and it does not reflect the actual length.

4.3.2 Silence Frame

A node transmits the silence frame when it does not want to transmit the token to another node at the end of the frame or when no other kind of frames can be sent (or need to be sent). A silence frame may contain bursts with payload data. Upon transmitting a silence token, the node shall transmit a new frame.

For instance, the silence frame may be used by a HE node that is alone in the network. In that situation it shall send periodical access frames, and silence frames will cover the rest of the time. Another use is when the bursts to be transmitted do not fit in one single MPDU. In such a case it is possible to transmit several consecutive silence frames and a final data frame to pass the token to another node.

Upon reception of a silence token, no action needs to be done.

4.3.3 Channel Estimation Frame

Periodically, every node has to send a channel estimation frame so that nodes communicating with that node can estimate their channel and adjust their bit-loading table.

A channel estimation frame is composed by a token announce followed by a known pseudo-random sequence of 16 symbols. A channel estimation frame is not addressed to any specific node. Any node which receives a Channel Estimation frame shall consider this frame if the transmitter MAC address included in the Token Announce is part of a completed entry in the port solver table (see 9.1.4.1).

A channel estimation process may lead to the transmission of a bit-loading table calculated upon the reception of a Channel Estimation Frame. The Adaptive Bit-Loading Protocol is used for the transfer of this bit-loading table and its acknowledgement (see section 9.1.2). Every node is responsible for evaluating the appropriate time to return the calculated bit-loading table. For instance, if there are major changes in the new calculated bit-loading table with respect to the previous one, the node can immediately return the new bit-loading table before any other transmission is considered. If the new bit-loading table has undergone no changes, the node shall not return any bit-loading table.

4.3.3.1 Transmission of channel estimation frames by TDRs and CPEs

TDRs and CPEs shall ask permission to their master to transmit a channel estimation frame. The channel estimation frame sent by a TDR may be used by the master and the slaves of the TDR to estimate the channel from the TDR. The channel estimation frame sent by a CPE may be used by the master of the CPE to estimate the channel from the CPE. A TDR shall not transmit channel estimation frames within its validity period unless the *Allow channel estimation frame* field received within the data token was set to 1 by its master, but can grant rights to transmit channel estimation frames to its slaves.

As shown in Figure 26, the channel estimation process can be initiated either by the TDR or CPE (referred as slaves) or its master.

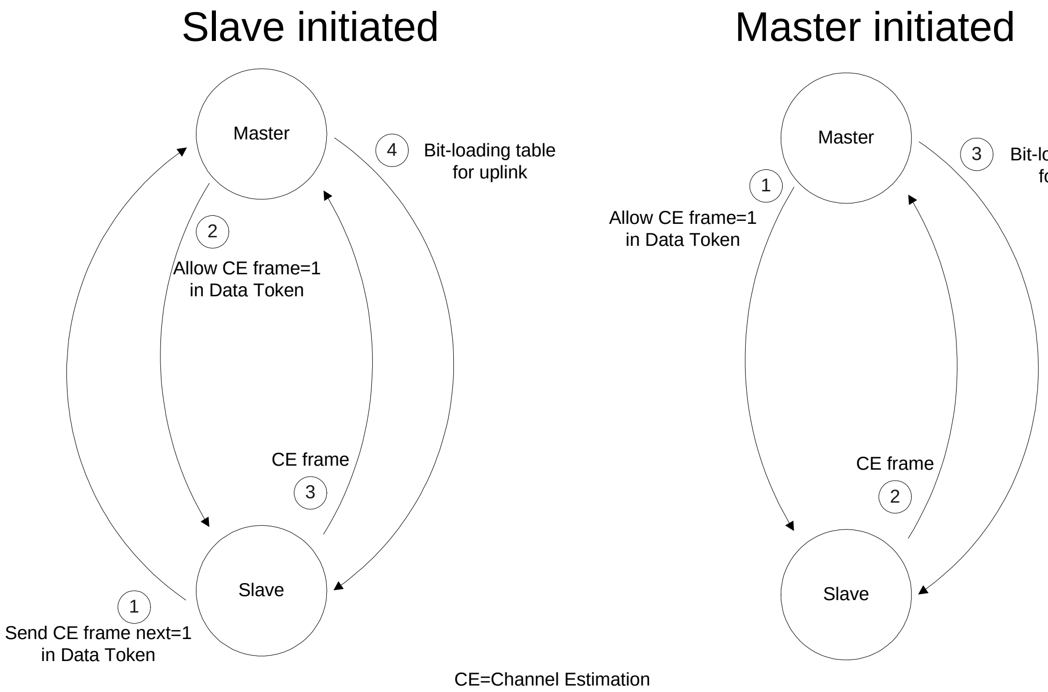

Figure 26 Transmission of channel estimation frames by TDRs and CPEs

⇨ Channel estimation initiated by the slave:

A slave shall periodically evaluate that it is time to send a new channel estimation frame. A slave shall not ask permission to send a channel estimation frame more than once every *MIN_INTERVAL_FOR_CE_REQUESTS* (default value is 3 seconds and corresponds to the minimum number of seconds between two requests from the same slave for an uplink channel estimation).

A slave may ask permission to send a channel estimation frame by setting the "*Send Channel Estimation Frame Next*" to 1 in the data token to its master. The master may accept or reject this request by setting the "*Allow Channel Estimation Frame*" to 1 (Accept) or 0 (Reject) in the next data token it returns to this slave. Upon reception of a data token with "*Allow Channel Estimation Frame*" set to 1, the slave shall transmit a Channel Estimation Frame. At the end of the transmission the slave shall enter reception mode, and the next transmitter will be the master.

⇨ Channel estimation initiated by the master:

The channel estimation process also allows the master for asking the slave to send a channel estimation frame. In this case, the master directly sets the "*Allow Channel Estimation*" to 1 in the data token sent to the slave. The process is then executed in the same way as mentioned above.

A master may request a slave to transmit a Channel Estimation frame as often as it deems necessary.

4.3.3.2 Transmission of channel estimation frames by the HE

The process for the HE is different since it is the only node in the PLC cell that does not have a master.

The HE can send a Channel Estimation Frame as often as it deems necessary. At the end of the transmission of a Channel Estimation frame by the HE, it is the HE which keeps the control of the medium.

The channel estimation frame sent by a HE may be used by the slaves of the HE to estimate the channel from the HE. Each slave will have the opportunity to return the new calculated bit-loading table (via the ABLP protocol) as soon as it receives a data token from its master. However, as described above, the slave shall not necessarily return the bit-loading table as soon as it receives a data token following a Channel Estimation frame.

4.3.4 Polling Frame

As described in 4.2.1, polling frames are used by the master to manage slave state transitions from Idle to Active or Idle to Unregistered.

Polling frames are frames terminated by a Polling Token. Polling Tokens are addressed to several slaves at the same time via the Polling Destination Bank/Polling Destination Ports fields of the Polling Token (see 4.4.2.3, Polling Token). A Polling Token is followed by a polling response sequence made of polling slots. The number of polling slots included in this sequence corresponds to the number of bits set to 1 in the Polling Destination Ports field of the polling token (max = 32). First bit set to 1 (going from the Least to the Most Significant bit) corresponds to the first polling slot in the polling response sequence and so forth. Upon reception of a polling token, a slave which is addressed by this token shall assert a SOT in its corresponding polling slot if that answer to the polling token is affirmative. It shall not respond if the answer to the polling token is negative.

Figure 27 shows a typical polling scheme in which three slaves are polled. Slave 1 and slave 3 assert a SOT in their corresponding polling slots (affirmative response) while Slave 2 does not respond to the polling token (negative response). Polling slots have duration equal to *Size Poll Window*. Two polling slots are separated by a gap of *Offset Poll Window*. The last polling slot of the Polling Response Sequence is not followed by this gap: the TX-RX Switch time shall be initiated immediately after the transmission of the last polling slot

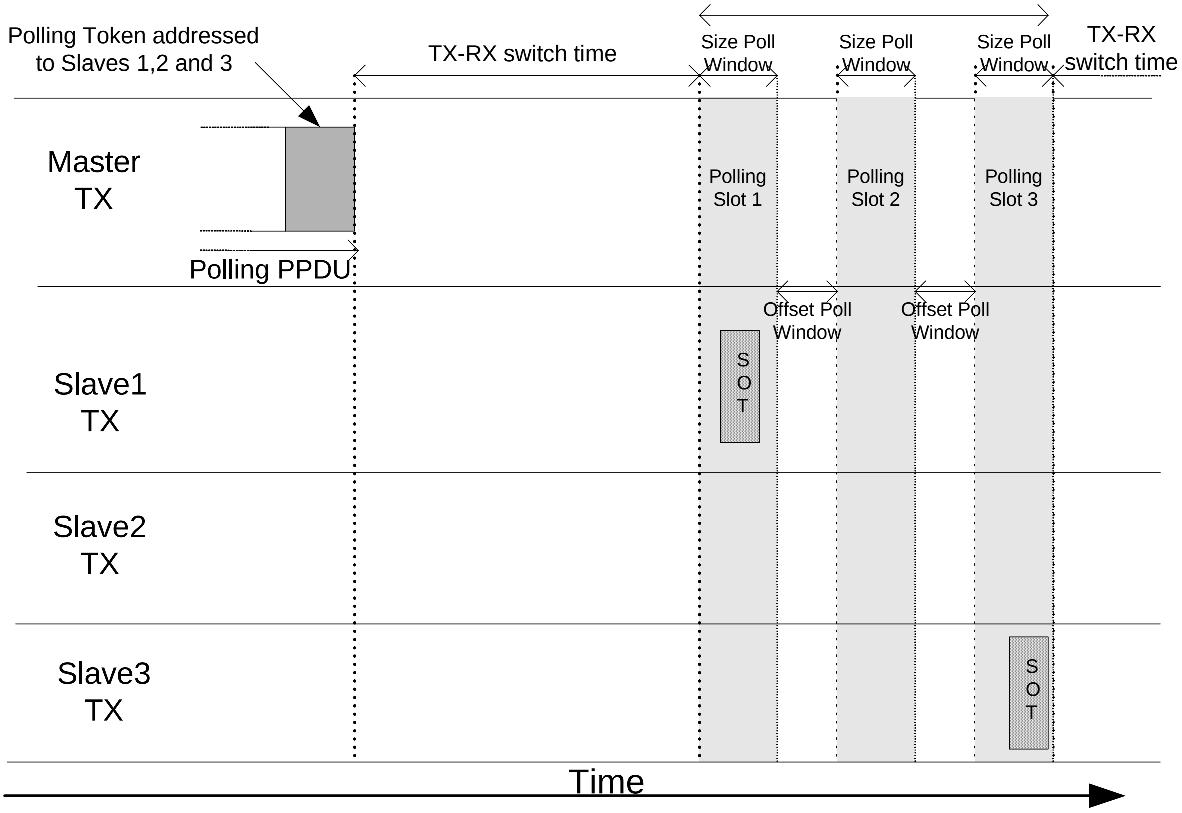

Figure 27 Polling scheme

As described in 4.2.1, there are two types of Polling Tokens transmitted by a master:

- Active Polling Token
- Alive Polling Token

These Polling Tokens are identified by the Polling Type field (see 4.4.2.3, Polling Token).

Note: Active Polling Tokens provide an Idle slave with the ability to recover the Active state and get transmission opportunities. The master transmits data tokens only to those slaves that are considered Active.

Alive Polling Tokens provide the master with the ability to detect slaves with no activity (e.g.: turned off). These ones become Unregistered and are not polled anymore.

4.3.5 Access Frame

A master shall periodically transmit Access tokens inviting Unregistered slaves or Registered slaves that want to change masters to subscribe to the network. An Access Token does not have a specific destination, and thus it is possible that no slave will answer to this request. Upon receiving an access frame, an Unregistered slave or a Registered slave that decides to reply must send an access reply frame as described in 4.3.6.

4.3.6 Access Reply Frame

The access reply frame is sent by an Unregistered slave that wishes to enter the network, or a Registered slave that wishes to change its current master. The transmission of an Access Reply Frame is triggered by the reception of an Access Frame.

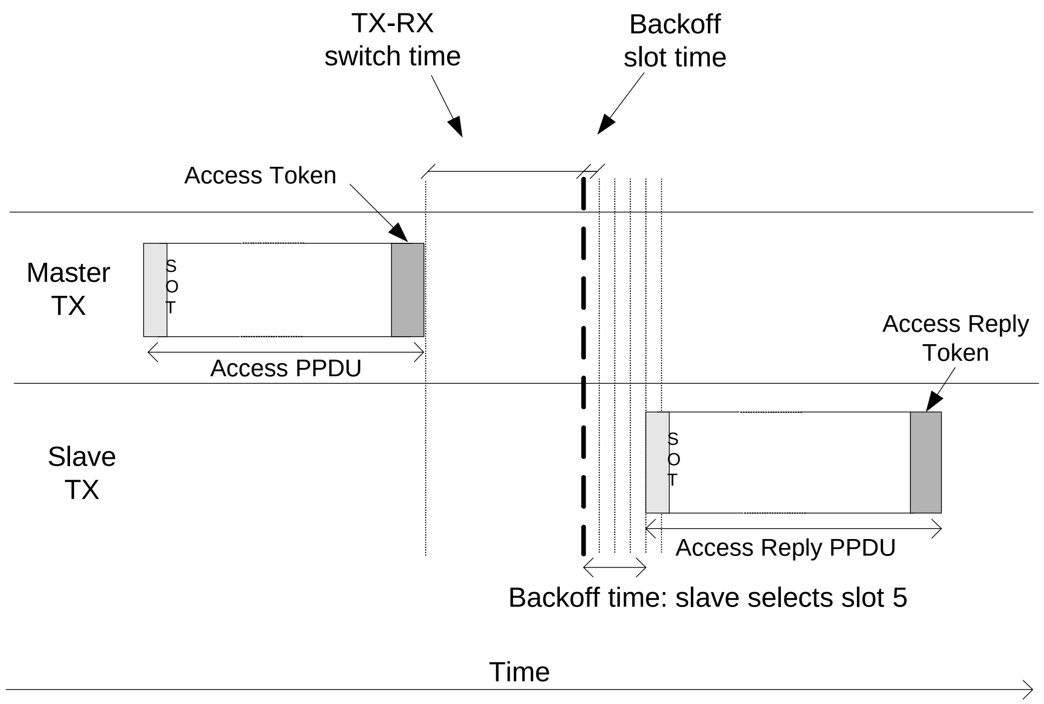

Figure 28 Access Reply scheme

The Access Reply Frame is preceded by a contention. The contention method used is the backoff contention method: all contending nodes select a random amount of time to wait before beginning the transmission. The SOT can be transmitted in one of 16 slots. Each node that intends to send an Access Reply PPDU will select a random number between 1 and 16 with a uniform distribution and schedule the transmission of the PPDU in the appropriate slot (slot 1 corresponds to the first slot after the TX-RX switch time has expired). The size of each slot is *BACKOFF_SLOT_TIME*. After the backoff time has expired, the Access Reply PPDU must be transmitted. If a SOT (beginning of an Access Reply PPDU) is heard before the time has expired, another node has won the contention and the transmission of the access reply frame should be deferred until a new access frame is received. A new random slot will be selected the next time. Particularly, the backoff process is not affected by the loss of a contention, and the process will be carried out in the same manner, except that the random slot will be selected again.

If no SOT is heard before starting the transmission of the own SOT, the whole transmission of the access reply frame will take place. Its structure is identical to a data frame, except for the final token type.

The master, upon receiving an access reply frame, will consider, taking into account Radius, QoS or any other criteria defined, accepting this node as its slave. The response will be sent as an access packet (see 9.1.3.1). If a node that transmitted an Access Reply PPDU does not receive a reply from the master in *ACCEPTATION_TO* it will begin the process again.

4.3.7 Non-returnable Data Frame

The non-returnable data frame is a frame transmitted from a master to many (up to eight) of its slaves (usually TDRs). The non-returnable data token transmitted at the end of the frame indicates which are the destination nodes and for how long they will be allowed to transmit. This length of time is known as the token validity, and is expressed as a number of symbols. The non-returnable data frame can only be sent by a master if Cluster Discovery Protocol (0) messages have been received from the TDRs that are slaves to it.

Upon transmitting a non-returnable data token, the master will ignore any signals received until the validity time is finished. At this point, the master will transmit a new frame. No frame will be expected by the transmitter of the non-returnable data frame.

Upon reception of the non-returnable data frame, the slaves (usually TDRs) will start transmitting new frames. However, they must not transmit any data in the upstream (data for the master that sent the non-returnable data frame), or any channel estimation frames.

The non-returnable data frame allows the simultaneous transmission of data and tokens of several isolated clusters, thus allowing for spatial reuse.

4.4 MAC delimiters

- As described in 4.3, the MAC layer generates two types of MPDUs: regular MPDUs and Channel Estimation MPDUs. These MPDUs make use of two MAC delimiters:
- Token Announce
- Token

Note: MAC delimiters are processed by the PHY layer as described in 3.3.1.

4.4.1 Token announce

The Token Announce is sent in the initial sequence of every transmission frame. The Token Announce has the following fields:

- *Preamble* – 32 bits. The preamble is equal to 0x90471D2D. The preamble is used to distinguish the different types of control symbols.

- *Type* – 4 bits. In this case, *Type* should be ANNOUNCE(3).

- *Validity* – 12 bits. Maximum number of symbols that will be transmitted in the current transmission after this Token Announce Symbol. The maximum number that can be used is *MAX_TA_VALIDITY*

- *Access Id* – 16 bits. This id must be set to 0x0001.

- *Channel Estimation Frame* – 1 bit. If this frame is a Channel Estimation Frame, *Channel Estimation Frame* must be set to 1. Otherwise, *Channel Estimation Frame* must be set to 0.

- *Power Control bits* – 5 bits. From left to right:

    - Bit 4. Set to 1 on booting and after the transmission gain is increased. Set to 0 fifteen seconds after it was set to 1.

    - Bit 3. Set to 1 if the node cannot decrease its transmission gain due to own reasons. For example: low throughput. Set to 0 otherwise.

    - Bit 2. Set to 1 if the node is not going to decrease its transmission gain due to other nodes reasons, but if those external circumstances end then the transmission gain could be decreased (depending on bit 3). For example: when another equipment has bit 3 to 1 and the node wants to isolate from it. Set to 0 otherwise.

    - Bit 1. Set to 1 if the power control task is on. Set to 0 otherwise.

    - Bit 0. Set to 1 if transmission gain has been decreased and isolation is expected. Set to 0 if transmission gain has its maximum (initial) value.

    If power control is not implemented this field shall be set to 0b01000

- *MAC Mode* – 3 bits. Must be set to 0b001 for Access.

- *Type of Node* – 3 bits. See below

- *ACCRQ* – 1 bit. If this frame is an Access Frame, *ACCRQ* must be set to 1. Otherwise, *ACCRQ* must be set to 0. Using this field, a node that is searching for a master only needs to receive the TA to know that it is an Access frame.

- MID – 21 bits. Master Id. It is used to detect if there is interference from other PLC cell. The problem of interference appears when there is more than one node allocating the channel in time. This can happen when there are two PLC cells, each one having its own HE, and the transmissions from some nodes in one of them can be received by some nodes in the other PLC cell.

  The first problem is to know if a node is being interfered because there is more than one HE in the channel. To solve this issue all the nodes that share the token must identify themselves as members of the same PLC cell.

  The HE of the PLC cell must transmit an identifier, so called Master Id [MID], and each node that depends directly or indirectly (because is in the HE's PLC cell) must copy the MID.

Figure 29 Master Id

  - Force master: A code to force the priority of the MID (explained below)
    - For MV Nodes
      - 0b00011
      - 0b00111
    - For LV Nodes
      - 0b01011
      - 0b01111
      - 0b10011
  - Identifier: In this field the last two bytes of the MAC are copied but in reverse order. This field is used to avoid matching the MID between two different masters.
    Example: A master with MAC=0x0050C22CF436, generates an Identifier=0x36F4.

  A node different from the HE has the default MID = 0x1FFFFF, which is invalid, until it registers in a master.

- *MAC* – 48 bits. This is the MAC address of transmitter.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th></th></tr>
<tr><td colspan="8">90</td><td rowspan="4">preamble</td></tr>
<tr><td colspan="8">47</td></tr>
<tr><td colspan="8">1D</td></tr>
<tr><td colspan="8">2D</td></tr>
<tr><td colspan="4">Type</td><td colspan="4">Validity [11:8]</td><td></td></tr>
<tr><td colspan="8">Validity [7:0]</td><td></td></tr>
<tr><td colspan="8">Access Id [15:8]</td><td></td></tr>
<tr><td colspan="8">Access Id [7:0]</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td>Ch. Est. Fr.</td><td>Always MUST be set to 0</td><td colspan="5">Power Control bits</td><td>Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="4">Always MUST be set to 0</td><td colspan="3">MAC Mode</td><td>TypeOfNode[2]</td><td></td></tr>
<tr><td colspan="2">Type of Node [1:0]</td><td colspan="5">MID [20:16]</td><td>ACCRQ</td><td></td></tr>
<tr><td colspan="8">MID [15:8]</td><td></td></tr>
<tr><td colspan="8">MID [7:0]</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="8">MAC[0]</td><td></td></tr>
<tr><td colspan="8">MAC[1]</td><td></td></tr>
<tr><td colspan="8">MAC[2]</td><td></td></tr>
<tr><td colspan="8">MAC[3]</td><td></td></tr>
<tr><td colspan="8">MAC[4]</td><td></td></tr>
<tr><td colspan="8">MAC[5]</td><td></td></tr>
</table>

Figure 30 Token Announce

The field *Type* has the following meanings:

3	ANNOUNCE

The field *Type of Node* has the following meanings:

0	HE
1	CPE
2	TDR

4.4.2 Token

There are six kinds of Token:

- Data Token
- Silence Token
- Polling Token
- Access Token
- Access Reply Token
- Non-returnable Data Token

The field *Type* has the following meanings:

0	DATA
2	NON-RETURNABLE
5	POLLING
11	SILENCE
13	ACCESS_REQUEST
14	ACCESS_REPLY

The field *Type of Node* has the following meanings:

0	HE
1	CPE
2	TDR

The field *Direction* has the following meaning:

0	DOWNSTREAM

1	UPSTREAM

4.4.2.1 Data token

The Data Token has the following fields:

- *Preamble* – 32 bits. The preamble is equal to 0x5F73EEE2.
- *Type* – 4 bits. *Type* shall be DATA (0)
- *Direction* – 1 bit. See above.
- *Transmission in HURTO* – 1 bit. Indicates whether the transmission from Token Transmitter to *Destination Port* is using HURTO or not.
- *Allow Channel Estimation Frame*– 1 bit. From Master to Slave. If it is set to 1, the Slave receiving this token must send a Channel Estimation Frame. See 4.3.3 for more information.
- *Send Channel Estimation Frame next* – 1 bit. From Slave to Master. The Slave is requesting to send a Channel Estimation Frame. See 4.3.3 for more information.
- *Single Validity* – 4 bits. It is expressed in slots (each slot is composed of 16 symbols). The Single Validity is an indication from the master on the validity that may be assigned for individual transmission. This information is only used as an example of how to distribute the token validity. However, the receiving node does not have to follow that distribution. In that case the receiving node may change the Single Validity to a new value, according to the validity that it assigns in the Data Token. Individual transmission means the transmission from a TDR to a CPE or a CPE to its TDR. This field is not used by CPEs. The relevant field for CPEs is the Validity. The Single Validity includes all symbols transmitted after the TA. The minimum Single Validity is 1.
- *Validity* – 12 bits. From Master to Slave. This indicates the maximum validity that the slave will be allowed to use before returning the token. It is expressed as a number of symbols. The master shall start a timer after the transmission of the Token Announce by the slave with the duration of the Validity it has assigned, and expect to receive the token back before this timer expires. If the slave is a repeater, the Validity includes the time needed for the repeater to communicate with its slaves.
- *Destination Port* – 7 bits. Local Port assigned to the intended receiver of the Data Token. See 9.1.4 for more information on the Port Solver protocol.

- *Frame Priority* – 8 bits. Only used from slave to master. The Frame Priority field in the token is used to convey the information of the priorities that have been transmitted within the frame, when the token is directed towards its master node. When the token is transmitted towards slave nodes, this field is not used. If bit X is equal to 1, there is at least one packet in this frame with priority X.

- *Unused frame symbols* (13 bits). Unused symbols in the current frame, compared to the Validity indicated in the token announce of the same transmission frame.

- *Received words* - 16 bits. The Received Words Field is used to communicate to the token receiver the number of 32-bit words that have been received from that destination since the last time, so that the bandwidth limitation algorithm can respond. Counting shall be performed over complete received packets when they are passed to the Convergence Layer.

- *Number of Downstream Nodes* – 8 bits From a CPE to its master, the Number of Downstream Nodes is equal to 0. From a TDR to its master, the Number of Downstream Nodes is equal to the number of nodes in its PLC subtree minus 1.

- *Maximum Number of Hops* – 4 bits. From a CPE to its master, the *Maximum Number of Hops* is equal to 0. A TDR will set the *Maximum Number of Hops* as the maximum value of number of hops amongst those received by the nodes hanging from it plus 1.

- *ABLP ACK*- 16 bits. This field is part of the ABLP protocol (9.1.2) The ACK information includes:

    - Bpcid: 8 bits, ABLP packet to be acknowledged (id extracted from the packet).

    - Port: 7 bits, local port with the ABLP receiver (see Figure 46). It may be different from the Destination port of the Data token.

The data token field (16 bits) is arranged in this way:

Bpcid	0	Local port

Figure 31 ABLP ACK Format

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th></th></tr>
<tr><td colspan="8">5F</td><td rowspan="4">preamble</td></tr>
<tr><td colspan="8">73</td></tr>
<tr><td colspan="8">EE</td></tr>
<tr><td colspan="8">E2</td></tr>
<tr><td colspan="4">Type</td><td>Direction</td><td colspan="3">Reserved</td><td></td></tr>
<tr><td colspan="5">Reserved</td><td>Transmission in HURTO</td><td>Alow Channel Estimation Fr.</td><td>Send Ch. Est. Fr. Next</td><td></td></tr>
<tr><td colspan="4">Single Validity</td><td colspan="4">Validity [11:8]</td><td></td></tr>
<tr><td colspan="8">Validity [7:0]</td><td></td></tr>
<tr><td>Reserved</td><td colspan="7">Destination Port</td><td></td></tr>
<tr><td colspan="8">Frame Priority</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="3">Always MUST be set to 0</td><td colspan="5">Unused Frame Symbols [12:8]</td><td></td></tr>
<tr><td colspan="8">Unused Frame Symbols [7:0]</td><td></td></tr>
<tr><td colspan="8">Received words [15:8]</td><td></td></tr>
<tr><td colspan="8">Received words [7:0]</td><td></td></tr>
<tr><td colspan="4">Always MUST be set to 0</td><td colspan="4">Maximum Number of Hops</td><td></td></tr>
<tr><td colspan="8">Number of Downstream Nodes</td><td></td></tr>
<tr><td colspan="8">ABLP ACK [15:8]</td><td></td></tr>
<tr><td colspan="8">ABLP ACK [7:0]</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
</table>

Figure 32 Data Token

4.4.2.2 Silence Token

The Silence Token includes the following fields of the data token:

- *Preamble* – 32 bits. The preamble is equal to 0x5F73EEE2.
- *Type* – 4 bits. *Type* shall be SILENCE (11).
- *Unused frame symbols* (13 bits). Unused symbols in the current frame, compared to the Validity indicated in the token announce of the same transmission frame.

7	6	5	4	3	2	1	0	
5F								preamble
73								preamble
EE								preamble
E2								preamble
Type				Always MUST be set to 0				
Always MUST be set to 0								
Always MUST be set to 0								
Always MUST be set to 0								
Always MUST be set to 0								
Always MUST be set to 0								
Always MUST be set to 0								
Always MUST be set to 0								
Always MUST be set to 0			Unused Frame Symbols [12:8]					
Unused Frame Symbols [7:0]								
Always MUST be set to 0								
Always MUST be set to 0								
Always MUST be set to 0								
Always MUST be set to 0								
Always MUST be set to 0								
Always MUST be set to 0								
Always MUST be set to 0								
Always MUST be set to 0								

Figure 33 Silence Token

4.4.2.3 Polling Token

The Polling Token has the following fields:

- *Preamble* – 32 bits. The preamble is equal to 0x5F73EEE2.
- *Type* – 4 bits. In this case, *Type* shall be POLLING (5).
- *Polling Type* – 1 bit. (The Polling Type could be equal to ACTIVE(1) or ALIVE(0). When *Polling Type* is ACTIVE, the Polling Token is requesting which nodes have data to be sent. When *Polling Type* is ALIVE, the Polling Token is requesting which nodes are alive.
- *Polling Destination Bank* – 2 bits. The total number of destination ports of this Polling Token is divided into 4 banks. The Polling Destination Bank is to know the destination bank of this polling. The first available port is port 9.

0 → 9 - 40 ports

1 → 41 - 72 ports

2 → 73 - 104 ports

3 → 105 - 126 ports

- *Unused frame symbols* (13 bits). Unused symbols in the current frame, compared to the Validity indicated in the token announce of the same transmission frame.

- *Polling Destination Ports* – 32 bits. Destination Ports of *Polling Destination Bank* of this Polling Token. Each bit represents one port.

  To cover all the possible ports, the field *Polling Destination Bank*, selects one of the four possible groups of ports. In order to poll port X (X>=9), *Polling Destination Bank* equals (X-9) / 32. The bit that represents port X in *Polling Destination Ports* field is bit (X-9) mod 32. A bit set to 1 means that this user is being polled. Only ports in the same Polling Destination Bank can be polled simultaneously

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th></th></tr>
<tr><td colspan="8">5F</td><td rowspan="4">preamble</td></tr>
<tr><td colspan="8">73</td></tr>
<tr><td colspan="8">EE</td></tr>
<tr><td colspan="8">E2</td></tr>
<tr><td colspan="4">Type</td><td>Polling Type</td><td colspan="2">Polling Destination Bank</td><td>Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td>1</td><td colspan="7">Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="3">Always MUST be set to 0</td><td colspan="5">Unused Frame Symbols [12:8]</td><td></td></tr>
<tr><td colspan="8">Unused Frame Symbols [7:0]</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="8">Polling Destination Ports</td><td></td></tr>
<tr><td colspan="8">Polling Destination Ports</td><td></td></tr>
<tr><td colspan="8">Polling Destination Ports</td><td></td></tr>
<tr><td colspan="8">Polling Destination Ports</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
</table>

Figure 34 Polling Token

4.4.2.4 Access Token

The Access Token has the following fields:

- *Preamble* – 32 bits. The preamble is equal to 0x5F73EEE2.
- *Type* – 4 bits. Type shall be ACCESS_REQUEST (13).
- *Direction* –1 bit. Shall be set to DOWNSTREAM (0)
- *Type of Node* – 2 bits. See above.
- *Validity* – 12 bits. From Master to Slave. This indicates the maximum validity that the slave will be allowed to use before returning the token. It is expressed as a number of symbols.
- *Number of Hops* – 4 bits. For a Head End, the *Number of Hops* is equal to 0. For a TDR, the Number of Hops is equal to the *Number of Hops* read in Access Token from its Master + 1.
- *Unused frame symbols* (13 bits). Unused symbols in the current frame, compared to the Validity indicated in the token announce of the same transmission frame.
- *MAC Address* – 48 bits. Own MAC address.
- *Equivalent speed* – 8 bits. This value gives an indication of the throughput to the backbone that can be achieved connecting to the network through this node. The value is calculated from the equivalent BPS.

$$Equivalent\_speed = \left\lfloor \frac{BPS_{eq} - 120}{60} \right\rfloor$$

Equation 33

If the node is directly linked to the backbone the field will be set to 255. If the node is one hop from the backbone the equivalent BPS are equal to the average of the downstream and upstream BPS of the node with its master.

$$BPS_{eq} = \frac{BPS_{up} + BPS_{down}}{2} = BPS_{eq}^{own}$$

Equation 34

If there is more than one hop to the backbone the equivalent BPS are calculated from the BPS value obtained from the master and the own equivalent BPS.

$$BPS_{eq}^{i} = \frac{1}{\frac{1}{BPS_{eq}^{i-1}} + \frac{1}{BPS_{eq}^{own}}}$$

Equation 35

Values for the equivalent speed obtained from the previous formulas shall be between 1 and 254. A value of zero in this field means that the repeater has not been able to calculate it. The BPS information obtained from the access token, combined with the BPS calculated by the new slave in the link may be used in order to decide which is the best master to be connected to.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th></th></tr>
<tr><td colspan="8">5F</td><td rowspan="4">preamble</td></tr>
<tr><td colspan="8">73</td></tr>
<tr><td colspan="8">EE</td></tr>
<tr><td colspan="8">E2</td></tr>
<tr><td colspan="4">Type</td><td>Direction</td><td colspan="2">Type of Node</td><td>Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="4">Always MUST be set to 0</td><td colspan="4">Validity [11:8]</td><td></td></tr>
<tr><td colspan="8">Validity [7:0]</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
<tr><td colspan="4">Always MUST be set to 0</td><td colspan="4">Number of Hops</td><td></td></tr>
<tr><td colspan="3">Always MUST be set to 0</td><td colspan="5">Unused Frame Symbols [12:8]</td><td></td></tr>
<tr><td colspan="8">Unused Frame Symbols [7:0]</td><td></td></tr>
<tr><td colspan="8">MAC [0]</td><td></td></tr>
<tr><td colspan="8">MAC [1]</td><td></td></tr>
<tr><td colspan="8">MAC [2]</td><td></td></tr>
<tr><td colspan="8">MAC [3]</td><td></td></tr>
<tr><td colspan="8">MAC [4]</td><td></td></tr>
<tr><td colspan="8">MAC [5]</td><td></td></tr>
<tr><td colspan="8">Equivalent Speed</td><td></td></tr>
<tr><td colspan="8">Always MUST be set to 0</td><td></td></tr>
</table>

Figure 35 Access Token

4.4.2.5 Access Reply Token

The Access Token has the following fields:

- *Preamble* – 32 bits. The preamble is equal to 0x5F73EEE2.

- *Type* – 4 bits. Type shall be ACCESS_REPLY (14).
- *Direction* –1 bit. Shall be set to UPSTREAM (1).
- *Type of Node* – 2 bits. See above.
- *Destination Port* – 7 bits. Local Port assigned to the master.
- *Number of Hops* – 4 bits. For a Head End, the *Number of Hops* is equal to 0. For a TDR, the Number of Hops is equal to the *Number of Hops* read in Access Token from its Master + 1. In a CPE, this field is not used.
- *Unused frame symbols* (13 bits). Unused symbols in the current frame, compared to the Validity indicated in the token announce of the same transmission frame.
- *MAC Address* – 48 bits. The MAC address of the master the access reply token is addressed to.

7	6	5	4	3	2	1	0	
5F								preamble
73								preamble
EE								preamble
E2								preamble
Type				Direction	Type of Node		Always MUST be set to 0	
Always MUST be set to 0								
Always MUST be set to 0								
Always MUST be set to 0								
Always MUST be set to 0	Destination Port							
Always MUST be set to 0								
Always MUST be set to 0								
Always MUST be set to 0				Number of Hops				
Always MUST be set to 0			Unused Frame Symbols [12:8]					
Unused Frame Symbols [7:0]								
MAC[0]								
MAC[1]								
MAC [2]								
MAC [3]								
MAC [4]								
MAC [5]								
Always MUST be set to 0								
Always MUST be set to 0								

Figure 36 Access Reply Token

4.4.2.6 Non-returnable data token

The Non-returnable Data Token has the following fields:

- *Preamble* – 32 bits. The preamble is equal to 0x5F73EEE2.
- *Type* – 4 bits. *Type* shall be NON-RETURNABLE (2)
- *Single Validity* – 4 bits. See description in 4.4.2.1
- *Validity* – 12 bits. Same meaning as in the Data Token (4.4.2.1). Since this token is non-returnable, the validity will be fully used.
- *Destination Port 0 to 7* – 8x7 bits. Local Port of the intended receiver of the Data Token. If Destination Port X is not used, it shall be set to 0. At least one of the destination ports shall be different to 0.

The implementation of this type of token is optional.

7	6	5	4	3	2	1	0	
5F								preamble
73								preamble
EE								preamble
E2								preamble
Type				Always MUST be set to 0				
Always MUST be set to 0	Destination Port 0							
Single Validity				Validity [11:8]				
Validity [7:0]								
Always MUST be set to 0	Destination Port 1							
Always MUST be set to 0								
Always MUST be set to 0	Destination Port 2							
Always MUST be set to 0	Destination Port 3							
Always MUST be set to 0			Unused Frame Symbols [12:8]					
Unused Frame Symbols [7:0]								
Always MUST be set to 0	Destination Port 4							
Always MUST be set to 0	Destination Port 5							
Always MUST be set to 0	Destination Port 6							
Always MUST be set to 0	Destination Port 7							
Always MUST be set to 0								
Always MUST be set to 0								
Always MUST be set to 0								
Always MUST be set to 0								

Figure 37 Non-returnable Data Token

4.5 Token Loss Recovery

A token loss occurs when the master node transmits a frame and the intended token receiver does not detect the token or when the master does not detect the token sent by one of its slaves. In case of token loss, the master node shall detect that the token has been lost.

Using the validity associated to every transmitted token, the master node shall generate a new frame whenever the validity assigned in a token to one of its slaves expires without receiving back the token. Only HE and TDR implement this feature.

4.6 MAC parameter specification

Table 10 contains the values for the MAC parameters.

Parameter	Value	Description
Number of channel estimation symbols per frame	16	See 4.3.3
Maximum number of simultaneous polling users	32	Maximum number of users that can be polled with one single polling frame. See 4.3.4
Offset poll windows	61.2 us	See 4.3.4
Size poll window	81.2 us	See 4.3.4
MIN_TA_VALIDITY	1 Symbol	See 4.4.1
MAX_TA_VALIDITY	240 Symbols	See 4.4.1
MIN_TOKEN_VALIDITY	16 symbols	See 4.4.2.1
MAX_TOKEN_VALIDITY	4095 symbols	See 4.4.2.1
BACKOFF_SLOT_TIME	35.625 us	See 4.3.6
Backoff maximum number of slots	16	See 4.3.6
MAX_ACTIVE_POLL_INTERVAL	2 seconds	See 4.2
MAX_ALIVE_TOKENS	100 tokens	See 4.2

MAX_ALIVE_POLL_INTERVAL	5 seconds	See 4.2
MAX_ACCESS_INTERVAL	5 seconds	See 4.2
Channel Estimation Maximum Period	10 seconds	Channel Estimation Frame should be sent at least every *Channel Estimation Maximum Period*.
ACCEPTATION_TO	5 seconds	See 4.3.6

Table 10 MAC parameter specification

5 LLC

5.1 Introduction

The LLC layer receives packets (Convergence Layer PDUs) from the Convergence Layer and transmits LLC PDUs to the MAC layer. LLC tasks are:

- Segmentation and grouping of packets to create bursts payloads.
- The addition of an LLC delimiter (burst header) to every burst payload.
- The Transfer Control Protocol for sequence control and acknowledgement of bursts.

5.2 Burst header

The Burst Header is the LLC Delimiter which precedes every burst. It has the following fields:

- *Preamble* – 32 bits. Always set to 0xACBCD211.
- *Transmission Port* – 7 bits. Local Port of the intended receiver of the Burst.
- *APL* – 14 bits. Length of the payload in 32-bit words of the current burst, including RS FEC redundancy. The APL will be defined by the RS FEC redundancy used and the length of the data included in the burst.
- *FEC payload* – 6 bits. RS FEC codeword payload length in 32-bit words. The RS FEC codeword payload allowed values are indicated in Table 2. The FEC Redundancy + FEC payload has to be less or equal to 63 words.
- *FEC Redundancy* – 2 bits. RS FEC codeword redundancy length:
    - *0*: 8 redundancy bytes (2 words)
    - *1*: 12 redundancy bytes (3 words)
    - *2*: 16 redundancy bytes (4 words)
    - *3*: 20 redundancy bytes (5 words)
- *BPS* –14 bits. Bits per symbol for the current burst. Depends on the encoding used for the current transmission. Its values range from 24 to 14592. The BPS calculation (Equation 17) includes the RS redundancy and excludes the 4D-TCM. With the BPS

and APL values nodes which are not the destination of the burst, but may be the destination of subsequent bursts in the same frame, may calculate the burst duration in symbols in order to know when the next burst header or token will be placed.

- *Burst Id* – 5 bits. Identification of this burst. The Burst Id numbering is independent for each Local Port. It shall be increased by one for each burst that is sent to the same Local Port.
- *HURTO* – 1 bit. Will be set to 0b1 when the burst is transmitted in HURTO mode.
- *Dummy Hdr*– 1 bit. It shall be set to 1 if the burst is not carrying data symbols.
- *Crypto* – 1 bit. Shall be set to 0b1 when the burst is encrypted.
- *ACK Info* – 1 bit. Shall be set to 0b1 when the burst header contains Transfer Control protocol information (see 5.4).
- *ACK Id* – 5 bits. Burst identification of the last correctly received burst.
- *ACK En* – 1 bit. Shall be set to 0b1 when the reception node of this burst must generate an ACK for it.
- *ACK Port* – 7 bits. Local Port over which the ACK contained in the header takes effect.
- *Priority* – 8 bits. Indicates the priorities belonging to the packets contained in this burst. If bit X is equal to 1, there is at least one packet in this burst with priority X.
- *Cipher Key* – 64 bits. See 10.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th></th></tr>
<tr><td colspan="8">AC</td><td rowspan="4">preamble</td></tr>
<tr><td colspan="8">BC</td></tr>
<tr><td colspan="8">D2</td></tr>
<tr><td colspan="8">11</td></tr>
<tr><td colspan="3">0</td><td colspan="5">Transmission Port [6:2]</td><td></td></tr>
<tr><td colspan="2">Transmission Port [1:0]</td><td colspan="6">APL [13:8]</td><td></td></tr>
<tr><td colspan="8">APL [7:0]</td><td></td></tr>
<tr><td colspan="6">FEC Payload</td><td colspan="2">FEC Redundancy</td><td></td></tr>
<tr><td colspan="8">BPS [13:6]</td><td></td></tr>
<tr><td colspan="6">BPS [5:0]</td><td colspan="2">BURST ID [4:3]</td><td></td></tr>
<tr><td colspan="3">BURST ID [2:0]</td><td>HURTO</td><td>DUMMY Hdr</td><td>Crypto</td><td>ACK Info</td><td>ACK ID [4]</td><td></td></tr>
<tr><td colspan="4">ACK ID [3:0]</td><td colspan="4">Always MUST be set to 0</td><td></td></tr>
<tr><td>ACK En</td><td colspan="7">ACK Port</td><td></td></tr>
<tr><td colspan="8">PRIORITY</td><td></td></tr>
<tr><td colspan="8">Cipher Key [63:56]</td><td></td></tr>
<tr><td colspan="8">Cipher Key [55:48]</td><td></td></tr>
<tr><td colspan="8">Cipher Key [47:40]</td><td></td></tr>
<tr><td colspan="8">Cipher Key [39:32]</td><td></td></tr>
<tr><td colspan="8">Cipher Key [31:24]</td><td></td></tr>
<tr><td colspan="8">Cipher Key [23:16]</td><td></td></tr>
<tr><td colspan="8">Cipher Key [15:8]</td><td></td></tr>
<tr><td colspan="8">Cipher Key [7:0]</td><td></td></tr>
</table>

Figure 38 Burst Header

5.3 Burst structure

A burst is composed of a Burst Header delimiter followed by a data payload including one or several fragmented and/or completed packets. A Burst Header delimiter without any following data payload is used to send ACK when there are no data to be sent.

Note: The Burst Header is a delimiter processed by the PHY layer as described in 3.3.1. It is transmitted in HURTO mode over one single symbol. The data payload is processed by the PHY layer as described in 3.3.2. The data payload may use HURTO or adaptive mapping. It can be transmitted over one or several symbols.

The Burst could have the following structures:

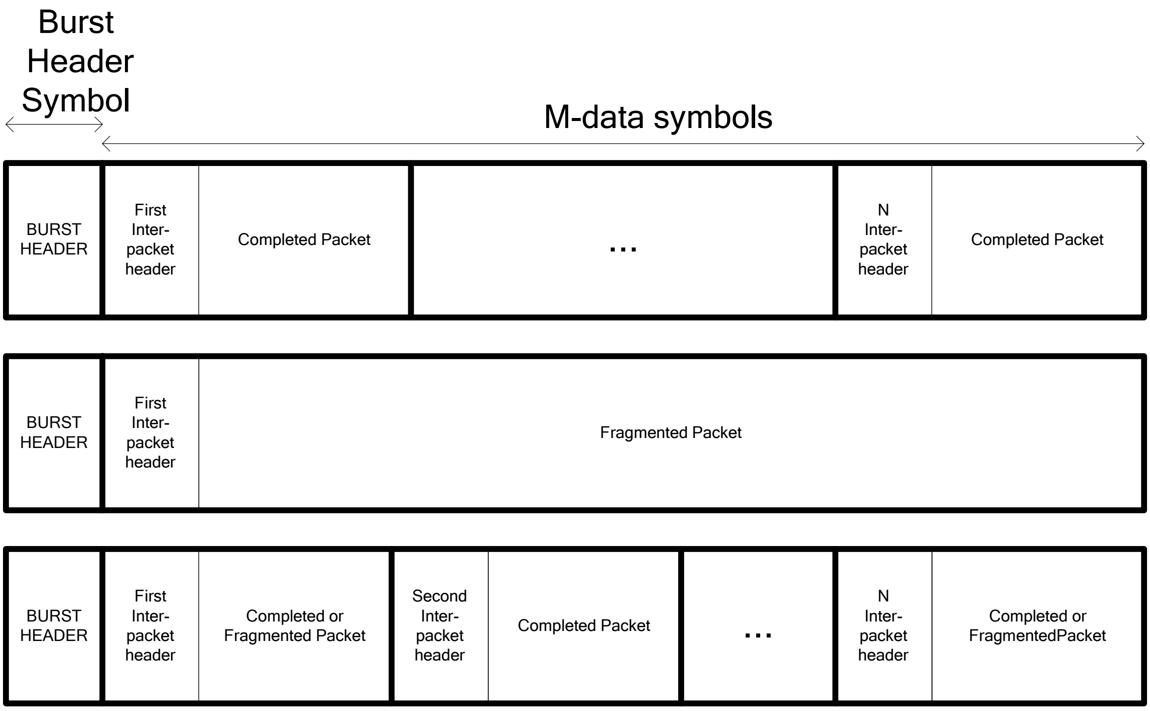

Figure 39 Burst examples

For more information about the Burst Header see 5.2.

Each burst can contain more than one OPERA packet. For more information about the OPERA packet format see 6.2.

The length of a burst is equal to a number of data symbols. The maximum length of a burst is 30 OFDM symbols, excluding the burst header. The maximum capacity of the burst in 32-bit words depends on the bit-loading table and the number of data symbols.

When one packet has to be divided into fragments, each fragment length must be multiple of 256 bytes, except the last fragment. No 32-bit words are added as padding in the last fragment to obtain 256 bytes. If one packet has length equal to X with X = Y * 256 + Z bytes, the packet could be sent in up to (Y+1) different bursts using fragmentation. If one packet has length equal to X bytes with X less than 256 bytes, this packet is sent with X bytes plus the needed padding to align it to 32-bit words. In any case, the only added padding is to be aligned to 32-bit words.

The inter-packet header is used for disassembling bursts in reception. An inter-packet header shall always precede each packet or fragment of a packet.

There are several restrictions regarding fragmentation of packets. Packet fragments shall only appear at the beginning or the end of the burst. Once a fragment of a packet is sent in a burst, the rest of the packet shall be sent before any other packet to the same Local Port (regardless of priority). A node shall not fragment packets to a given Local Port if a maximum length OPERA packet (1536 bytes) fits in 10 OFDM symbols or less.

Packets with the same destination port and same priority shall be transmitted in order. Within the same destination port, packets are transmitted according to their priority, with highest priority packets first (see 8.2.1).

The inter-packet header has the following structure:

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th>Octet</th></tr>
<tr><td colspan="8">0</td><td>1</td></tr>
<tr><td>Part of Packet</td><td>Unfinished</td><td>Burst Change Mode</td><td>0</td><td colspan="4">Number of Packets [8:5]</td><td>2</td></tr>
<tr><td colspan="5">Number of Packets [4:0]</td><td colspan="3">Length [8:6]</td><td>3</td></tr>
<tr><td colspan="6">Length [5:0]</td><td colspan="2">Padding Last Word</td><td>4</td></tr>
</table>

Figure 40 Inter-packet header

- Part of Packet – 1 bit. The first fragment in this burst is part of a previous packet. This field shall only be checked by the receiver in the first inter-packet header.

- Unfinished – 1 bit. The last fragment in this burst is unfinished. This field shall only checked by the receiver in the first inter-packet header.

- Burst Change Mode – 1 bit. It is the first burst from this source in Transfer Control Protocol equal to Mode ACK. This is the first burst from this source or the previous bursts were in Transfer Control Protocol equal to Mode no ACK. This field shall only checked by the receiver in the first inter-packet header. See 5.4 for more information.

- Number of packets – 9 bits. Number of packets or fragments in the burst after this inter-packet header. For example, if a burst contains 10 packets, this field will be set to 10 in the first inter-packet header, it will be set to 9 in the second, and so on.

- Length – 9 bits. Length of the packet or fragment in words.

- Padding Last Word – 2 bits. Number of padding bytes of the last word of the packet or fragment of a packet.

5.4 Transfer Control Protocol

The Transfer Control protocol has the following functions:

1) Control of sequence of received bursts in the receiver

2) Acknowledge of last correctly received burst, which implies request of retransmission of frames received with errors.

The Transfer Control protocol may be used in two modes:

a) No ACK mode: Only control of sequence is used. With this configuration, the receiver cannot request any retransmission.

b) ACK mode: Control of sequence and acknowledge (ACK) of bursts.

The Transfer Control mode of one connection is independent from that of the other connection. This is, the connection from node A to B may use different Transfer Control protocol than connection from node B to A, and so on with relation to other connections.

The Transfer Control fields are in the burst header or in the inter-packet header and are the following:

- *ACK Info* - 1 bit. Active when the header contains an ACK of another burst
- *ACK Id* – 5 bits: ACK identification: last correctly received burst id.
- *ACK Port* – 7 bits: Destination port of this ACK
- *ACK En* – 1 bit: Set to zero if the current burst is using No ACK mode and set to one if the burst is using ACK mode.
- *Burst Id* – 5 bits: Burst identification
- *Burst Change Mode* – 1 bit:
    - o Set to 0b1 (CHANGE) if the previously sent burst was using No ACK and this burst is using ACK.

Set to zero (NO CHANGE), if the previously sent burst was using the same mode than the current burst, or the previously sent burst was using ACK and the current burst is using No ACK.

5.4.1 No ACK mode

The No ACK mode allows the control of sequence in the receiver.

The following sequence of events occur in a connection between two nodes using No ACK mode:

1) Node A sends a burst to Node B and frees all packets inside this burst. This burst has *Burst Id* with a value of X, *ACK En* indicating No ACK and *Burst Change Mode* indicating NO CHANGE. The next burst to be transmitted from Node A to Node B must use *Burst Id* X+1.

2) Node B receives the burst with *Burst Id* X from Node A. The previous burst correctly received by Node B from Node A has a *Burst Id* with a value of Y.

    - If (Y+1) is equal to X, the received burst is assumed to be the expected burst. If Node B has one stored fragmented packet from burst with *Burst Id* Y, the received *burst is assumed to contain* the next fragment of the packet, and both fragments are concatenated. All complete packets in *Burst Id* X are sent to upper level.

    - If (Y+1) is different from X, the received burst is not the expected burst. If Node B has one stored fragmented packet from burst with *Burst Id* Y, this fragment cannot be concatenated with a fragment of the burst. So the stored fragment is discarded, the first fragment of the burst is discarded if the beginning of it is missing, and the following packets in the burst are processed. All complete packets extracted from the *Burst Id* X are sent to upper level.

5.4.2 ACK mode

The ACK mode allows the control of sequence in the receiver and the receiver can request retransmission of incorrectly received bursts.

The following sequence of events occur in the connection between two nodes using ACK protocol:

1) Node A sends a burst to Node B, with *Burst Id* equal to X, *ACK En* indicating ACK and *Burst Change Mode* indicating No change. The previous burst correctly received and accepted by Node B from Node A had *Burst Id* equal to Y. The next burst to be transmitted from Node A to Node B uses *Burst Id* X+1.

2) Node B receives the burst with *Burst Id* equal to X from Node A.

    - If (Y+1) is equal to X, the received burst is the expected burst. In this case, if Node B has one stored fragmented packet from burst with *Burst Id* Y, the received *Burst Id* X is assumed to contain the next fragment of this packet. All completed packets in *Burst Id* X are sent to upper level. An ACK with *ACK Id* equal to X and *ACK Port* equal to the port to communicate with Node A is sent to Node A.

- If (Y+1) is different from X, the received burst is not the expected burst. An ACK with *ACK Id* equal to Y and *ACK Port* equal to the port to communicate with Node A is sent to Node A. The burst with *Burst Id* X is discarded.

3) If node A receives the ACK with *ACK Id* equal to Y, the packets included in burst with *Burst Id* Y and less than Y are considered to be correctly received. The next *Burst Id* to be used in burst sent by Node A must be Y+1.

When the transmitter receives an ACK with *ACK Id* equal to Y different from last sent *Burst Id* equal to X, it means that the bursts with *Burst Id* from (Y+1) to X have to be retransmitted. The retransmitted bursts do not need to be equal to the transmitted bursts, but the transmitted packets shall be retransmitted in the same order and without inserting new packets.

5.4.3 Transition from ACK to No ACK

If Node B receives one burst with *ACK En* indicating ACK mode from Node A and the next burst has *ACK En* indicating No ACK from Node A, the receiver must proceed as in 5.4.1. No more ACKs must be sent from Node B to Node A unless the Node A sends a burst with ACK mode.

5.4.4 Transition from No ACK to ACK

If Node B receives one burst with *ACK En* indicating No ACK with *Burst Id* X from Node A, and the next burst with *ACK En* equal to ACK with *Burst Id* Y from Node A, there are the following possibilities:

- If the second burst has *Burst Change Mode* indicates CHANGE, the packets inside second burst are sent to upper levels and the Node B sends ACK with *ACK Id* Y to the Node A, regardless of the equality or inequality between Y and X+1.
    - If Y is equal to (X+1): if there are any stored fragment of packet from burst with *Burst Id* X, burst with *Burst Id* Y is assumed to contain the next fragment. If there are no errors, the Node B sends ACK with *ACK Id* Y to the Node A.
    - If Y is different from (X+1), if there is any fragmented packet from burst of packet received with *Burst Id* X, burst with *Burst Id* is assumed to not have any fragment of this stored fragmented packet, and then it is discarded.
- If the second burst has *Burst Change Mode* equal to NO CHANGE, the second burst is dropped. The second burst is dropped independently if Y is equal to or different from (X+1). The Node B does not request any retransmission.

If the burst with Burst Change Mode equal to CHANGE must be retransmitted, the Burst Change Mode must be equal to CHANGE again.

If in a Burst, the Burst Change Mode field is set to CHANGE, and the burst has been transmitted in ACK mode, and received correctly, the receiver must always:

- Accept the burst, provided it does not contain errors, independently of the Burst Id of the burst
- Reset the expected burst id for next burst to that of the received burst plus one, independently of the mode of the previously received burst.

5.4.5 Transmission of acknowledgements

If one node has ACK information to be sent and gains access to the channel, the ACK information must be sent inside the burst header. If this node cannot transmit data symbols, due to QoS or because there are no data to be transmitted, a Dummy Header is sent. A Dummy Header is a Header with data length equal to 0 and carrying ACK information. Because one node can receive from more than one node using ACK mode, this node could send more than one Dummy Header.

Every time that one header is sent, the ACK information may be included inside this header. For this reason, in the following situations the ACK has to be ignored:

1) The Node A receives ACK with *ACK port* identifying the communication with Node B but this communication uses No ACK mode.

2) The Node A receives ACK with *ACK port* identifying the communication with Node B. Node A is not waiting for the acknowledgement of any burst.

The ACK port must have a significant value because an acknowledgement can be sent in any header, addressed to any node, so the destination port of that header has no necessary relation to the ACK port.

In the following situations, an ACK is sent from the receiver of one burst to the sender of that burst.

- The *Burst Id* X with *ACK En* set to No ACK is received. If the following received burst has *Burst Id* Y with *ACK En* set to ACK and *Burst Change Mode* indicating CHANGE, the ACK with *ACK Id* equal to Y is sent from receiver of the burst to the transmitter of the burst. In this case, Y could be equal to or different from (X+1).
- The *Burst Id* X with *ACK En* set to ACK is received. If the following received burst has *Burst Id* Y with *ACK En* set to ACK and *Burst Change Mode* indicates CHANGE, the ACK with *ACK Id* equal to Y is sent from burst's receiver to burst's transmitter. In this case, Y could be equal to or different from (X+1).

- The *Burst Id* X with *ACK En* equal to ACK is received. If the next received burst has *Burst Id* Y with *ACK En* equal to ACK and *Burst Change Mode* indicates NO CHANGE, there are the following possibilities:

    - If Y is equal to (X+1), the ACK with *ACK Id* equal to Y is sent from burst's receiver to burst's transmitter.

    - If Y is different from (X+1), the ACK with *ACK Id* equal to X is sent from burst's receiver to burst's transmitter. The receiver is requesting retransmission of burst with Burst Id (X+1).

6 CONVERGENCE LAYER

6.1 Overview

This section describes the mechanisms that are used to encapsulate packets coming from external interfaces before passing them to the LLC. Some modifications are made to the Ethernet packet from the external interfaces. So the main task of the Convergence layer is Ethernet packet formatting.

6.2 Packet Formatting

The OPERA Packet Format is Ethernet-like packet format without the following fields:

- Preamble (PRE) - 7 bytes. The PRE is an alternating pattern of ones and zeros that tells receiving stations that a frame is coming, and that provides a means to synchronize the frame-reception portions of receiving physical layers with the incoming bit stream.

- Start-of-frame delimiter (SFD) - 1 byte. The SOF is an alternating pattern of ones and zeros, ending with two consecutive 1-bits indicating that the next bit is the left-most bit in the left-most byte of the destination address.

- Frame check sequence (FCS) - 4 bytes. This sequence contains a 32-bit cyclic redundancy check (CRC) value, which is created by the sending MAC and is recalculated by the receiving MAC to check for damaged frames.

See the following figure to understand the differences:

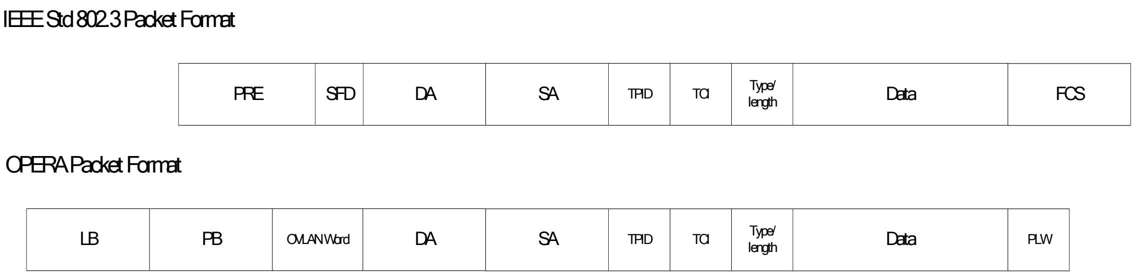

Figure 41 Packet format

The Packet Format has the following fields:

- LB – 48 bits. Last Bridge ID

- PB – 48 bits. Previous Bridge ID
- OVLAN – 32 bits with the following information:
  - PCF – 1 bit. Pre-classified Packet
  - PCP – 3 bit. Pre-classified Packet Priority
  - OVLANV - 1 bit.Valid OVLAN
  - OVLANID - 12 bits. OVLAN Identifier
- DA – 48 bits. Destination MAC Address. The DA field identifies which station(s) should receive the frame.
- SA – 48 bits. Source MAC Address. The SA identifies the sending station.
- TPID – 16 bits. Defined value of 8100 in hex. When a frame has the EtherType equal to 8100, this frame carries the tag IEEE 802.1Q/802.1p, otherwise it will be filled with 0x0000.
- TCI – 16 bits. VLAN Tag Control Information field including user priority, Canonical format indicator and VLAN ID. This field is always present, even if the TPID is not 0x8100.
- Len – 16 bits. IEEE Std 802.3-style Length/Type Field
- Data – From 42 to 1500 bytes. The Data Payload of this packet.
- PLW – From 0 to 3 bytes. Because the packet has to be 32 bit aligned, there are from 0 to 3 bytes of no valid data at the end of this packet. This field is called Padding Last Word.

The incoming IEEE std 802.3 packet may have TCI field or not, but the TCI field is always present in the OPERA packet format.

6.2.1 Maximum and Minimum Packet length

The maximum length of one packet is equal to 1536 octets.

The minimum length of one packet is equal to 76 octets.

6.2.2 OVLAN/VLAN Fields

The OVLAN has the same meaning of VLAN. The function is to provide IEEE 802.1Q functionality with an added feature in PLC nodes being able to define network architecture of standard IEEE 802.1Q and network architecture of PLC-VLAN (OVLAN). The purpose of

OVLANs is providing mechanisms for filtering Ethernet traffic at level 2 of the ISO/OSI layer, without requiring the use of 802.1Q VLANs. Ethernet filtering is required for two purposes:

- Avoiding Ethernet frames from one customer leaking into a neighbor customer (for security reasons).
- Avoiding Ethernet frames from one powerline sub-network (for example, an LV cell) leaking into another powerline sub-network (for security reasons and in order to avoid the problem of bridge tables filling up with unnecessary MAC addresses).

The motivation for avoiding the use of 802.1Q VLANs is that VLAN numbers are limited to less than 4096, and a large network can rapidly run out of VLANs numbers if they are used for customer isolation.

OVLAN filtering is optional in the system described by this standard but the OVLAN fields shall always be present in the packet.

The OVLAN fields are the following:

- OVLANV - 1 bit. Valid OVLAN
- OVLANID - 12 bits. OVLAN Identifier

The VLAN fields are the following:

- TPID – 16 bits. Tag Protocol Identifier
- TCI – 16 bits. Tag Control Information

The position of these fields inside packet format is described in 6.2.

Unlike IEEE 802.1Q standard, where the packet contains VLAN fields only if TCI has information, all packets in the system described by this standard must carry VLAN fields, with the following meaning:

- When TPID is different from 0x8100, the TCI is not used. This packet is not carrying valid 802.1Q information.
- When TPID is equal to 0x8100, the TCI has the same meaning than in IEEE 802.1Q.

The OVLAN fields have the following meaning:

- When OVLANV is equal to 0, the OVLANID is not used and the packet is not carrying OVLAN information.
- When OVLANV is equal to 1, the OVLANID is equal to the tag, and the tag is used with the same sense as in IEEE 802.1Q. The only difference is the following:

- The tag equal to 0 is a possible tag
- The tag equal to 0xFFF is a tag that must be accepted by all the filters.

Like 802.1Q where the TCI has priority information, the OVLAN word has two fields to know the priority of this packet:

- PCF: Pre-classified Packet
- PCP: Pre-classified Packet Priority

If PCF is set to 1, the priority of this packet is equal to PCP value. The 802.1p or any other rule is not applied to decide the priority.

In the case of nodes that do not perform OVLAN filtering, the OVLAN field will not be modified if the packet was received through a PLC interface. If the packet was received through other interfaces, the node will add the OVLAN fields with OVLANV = 1b0 and PCF = 1b0.

6.2.3 LB/PB Fields

These fields are used to avoid loops with broadcast packets.

Broadcast packets can be sent in two ways through the PLC network:

- Using the broadcast port. The broadcast packet is sent using the broadcast port and all nodes that have negotiated ports with the sender will receive it.
- Using unicast ports. The broadcast packet is sent once per each negotiated port, except the port where it was received in the case it was a PLC port.

In the case when a broadcast packet is sent using the broadcast port, these fields are used to avoid loops. In the other case, they are not necessary. In any case, they shall always be present in all packets.

The meaning of the fields (defined in 6.2) is the following:

- LB: Last Bridge ID (LB). It shall contain the MAC address of the node transmitting this packet.
- PB: Previous Bridge ID. It shall contain the MAC address of the previous node transmitting this packet.

When a packet enters the PLC network, Last Bridge ID shall be set to the current PLC node MAC address and Previous Bridge ID to 0.

In any retransmission, the Previous Bridge ID shall be filled with the Last Bridge ID value received. The Last Bridge ID shall be set to the current PLC node MAC address.

When receiving a packet it shall be discarded when its Previous Bridge ID is equal to the MAC address of node.

6.2.4 Relation between IEEE Std 802.3 and OPERA

The bytes from IEEE Std 802.3 are stored in 32 bits words in the following way:

Byte 3	Byte 2	Byte 1	Byte 0

Figure 42 Byte order

From 802.3 interface, the bytes are received in the following order: byte0, byte1, byte2 and byte3.

And from this 32 bits word, the bytes are transmitted into the LLC interface in the following order: byte3, byte2, byte1, byte0. The packets are transmitted to the LLC in the way shown in Figure 43, starting from the top, since the LLC works with bytes.

If we define N as the number of bytes in Data Payload plus the number of bytes of the Length/Type Field, then we can write N = 4*m + p; with m an integer value and p equal to 0, 1, 2 or 3. The number of bytes in PLW is equal to 0 if p is equal to 0 or 4-p in the other cases, because the packet is 32 bit aligned.

bits 7	6	5	4	3	2	1	0	Octet
LB[3]								1
LB[2]								2
LB[1]								3
LB[0]								4
PB[1]								5
PB[0]								6
LB[5]								7
LB[4]								8
PB[5]								9
PB[4]								10
PB[3]								11
PB[2]								12
OVLANID[7:0]								13
Reserved			OVLANV	OVLANID[11:8]				14
Reserved				PCP			PCF	15
Reserved								16
DA[3]								17
DA[2]								18
DA[1]								19
DA[0]								20
SA[1]								21
SA[0]								22
DA[5]								23
DA[4]								24
SA[5]								25
SA[4]								26
SA[3]								27
SA[2]								28
TCI[7:0]								29
TCI[15:8]								30
TPID[7:0]								31
TPID[15:8]								32
Data[1]								33
Data[0]								34
Lenght[0]								35
Length[1]								36
Data[5]								37
Data[4]								38
Data[3]								39
Data[2]								40
Data (from byte [6] to [4*m - 3])								
PLW (0 bytes if p is equal to 0 OR (4-p) bytes if p is different from 0)								
Last Word Data (from byte [4*m-2] to [4*m+p-3])								

Figure 43 Packet byte order to the LLC

6.3 Convergence layer service primitives

The bridging block communicates with the convergence layer through 3 service primitives Eth.req, Eth.cnf and Eth.ind:

- Eth.req (*802.3/EthernetII frame, PLC local ports list, PB, OVLAN word*): This primitive requests the transfer of an Ethernet frame from the local bridging block to a single peer bridging block or multiple peer bridging blocks in the case of multi-port transfer requests (see 6.3.1).

- Eth.ind (*802.3/EthernetII frame, PLC local port, LB, OVLAN word*): This primitive defines the transfer of an Ethernet frame from the convergence layer to the bridging block.

- Eth.cnf: This primitive is used by the convergence layer to notify the bridging block of the results of the Eth.req primitive.

Notes:

- VLAN tag may be included in the 802.3/EthernetII frame

- The OVLAN word includes PCF, PCP, OVLANV and OVLANID

6.3.1 Multi-port transfer requests

Multi-port transfer requests correspond to Eth.req service primitives including several PLC local ports as argument. Such requests are inherent to bridging in TDRs and HEs which manage several PLC ports. The multi-port requests are solicited by standard transparent bridging actions such as:

1. Broadcast transmission: The Ethernet frame includes a Broadcast Ethernet address in its Destination Address field. The multi-port request concerns all the PLC ports managed by the bridging block except the one from which the Ethernet frame was received (if it was eventually received from a PLC port).

2. Multicast transmission: The Ethernet frame includes a Multicast Ethernet address in its Destination Address field. The multi-port request might concern a subset of the PLC ports managed by the bridging block.

3. Bridge flooding: The bridging block could not find any association of a unicast Ethernet Destination addresses with a specific bridge port, the request is then flooded to all the bridge ports. The multi-port request concerns all the PLC ports managed by the bridging block.

6.4 Broadcast/Multicast handler functional requirements

The convergence layer shall include a broadcast/multicast handler that shall perform the following functions:

1. Handling transfer of multicast management messages: Management messages can be unicast or multicast as described in section 7.4. Multicast management messages can be received from the bridging block through primitives which are either single port requests or multi-port requests. Any multicast management message received through these requests shall be transferred over the broadcast port using HURTO modulation.

2. Management of multi-port transfer requests which do not carry multicast management messages: upon reception of such multi-port transfer request from the convergence layer, the block shall compute the best method for handling the transfer. If the multi-port request concerns all the local ports of the Port Solver Table or if the request includes a broadcast/multicast Ethernet Destination Address, depending on the bit-loading tables associated to each local port, the block might consider transferring the Ethernet frame over the broadcast local port using HURTO modulation. In the other cases, the multi-port request will result in several consecutive LLC service requests related to each unicast local port.

3. LB tagging: on the transmission path, the block shall set the LB field of the packet to the MAC address of the node if:

    a. the transmission includes a broadcast or multicast Ethernet frame;

    b. the transmission is performed over the broadcast port.

4. Broadcasts loop back avoidance: on the reception path, upon reception of a packet over the broadcast local port (127), the module shall discard any packet which is detected as previously transmitted (that is, the MAC address contained in the PB field of the packet is equal to the MAC address of the node).

5. On the reception path, upon reception of a packet over the broadcast local port (127), the local port passed within the Eth.ind primitive shall be the one assigned to the remote node from which the request was received.

7 BRIDGING REQUIREMENTS

7.1 Sending broadcast/multicast Ethernet frames

When sending a broadcast/multicast Ethernet frame or when flooding the ports of the bridge, the bridging block shall provide the list of PLC ports related to this transfer as part of the Eth.req service primitive.

7.2 Bridging between PLC ports

When bridging between PLC ports (HE or TDR feature), the bridging block shall return within the Eth.req a PB argument equal to the LB argument originally provided within the Eth.ind primitive. If PCF=1 within the OVLAN argument of the received Eth.ind, the PCF and PCP values (subfield of the OVLAN word argument) included in the subsequent Eth.req shall be kept unchanged.

7.3 Bridging from a non-PLC port to a PLC port

When bridging from a non PLC-port to a PLC port, the PB argument of the Eth.req shall be set to 0.

7.4 Filtering multicast management messages from a PLC port

Management messages are IEEE 802.2 SNAP encapsulated data units carried over an Ethernet frame which can be either unicast or multicast. Multicast management messages make use of one of the two following multicast addresses (referred as management multicast addresses): the STP multicast address 0x0180C2000000 (primarily used by the STP protocol but also used for some management messages) or the OPERA specific multicast address: 0x01139D000000. Any Ethernet frame including a management multicast address shall be filtered by the bridging block and not retransmitted to any other port. Upon reception of management messages destined to the above two mentioned multicast addresses, the bridging block shall provide these frames to the management port.

7.5 Filtering multicast management messages from the management port

Multicast management messages from the management port shall only be flooded to the PLC ports (non-PLC ports excluded).

8 QOS SERVICES

8.1 Introduction

This section describes mechanisms for applying specific guarantees in terms of bandwidth, latency and reliability to some flows. It is also possible to define a policy upon congestion. Any flow requiring such guarantees must initiate a connection through a Connection Admission Control procedure.

The mechanisms described here below are designed for flows circulating from the HE to a CPE or vice versa. These mechanisms have not been designed to support QOS for flows between CPEs,

The QOS services described here below rely on the configuration of several components:

1. Service classes. (must be declared in all the masters of a PLC cell).
2. Classifier module: this module maps flows onto service classes (preconfigured in all the nodes).
3. Congestion management (declared in all the masters).
4. Traffic specifications, published through the CAC procedure. Pre-determined traffic specifications must be declared in CPEs.

Note: A flow is a unidirectional data stream exchanged between two PLC units and carried over the same service class.

8.2 Service Class Definition

Up to eight service classes can be defined within a PLC cell. Services classes are globally defined over a PLC subcell. Four service parameters are necessary to fully describe a service class: priority, subcell_access_time, resource reservation type and service reliability.

8.2.1 Priority

Service classes are directly mapped onto priorities. A priority uniquely identifies a service class. That is why all the other service parameters are configured and mapped from this priority parameter. Priorities shall have values from 0 to 7, 0 being the lowest priority. If packets have to be dropped at the transmitter because of lack of resources, then the lower priority packets shall be dropped before the higher priority packets. When there are data of different priorities addressed to the same destination, higher priority data shall be transmitted earlier than lower priority data.

8.2.1.1 Configuration

Priority parameter is determined by the Classifier module operating at the bridging block level. The classifier analyses all the frames received from a bridge port (PLC port, non-PLC port, management port) and assigns a priority to each Ethernet frame transmitted to the convergence layer via the Eth.req primitive (see 6.3). The Classifier operates according to some predetermined and/or configurable rules (see Opera Specification: Part 2). Any frame received from the management port shall be mapped to priority 7 (that is: service class 7).

Note: For each Eth.req primitives delivered by the bridging block, a priority is submitted as part of the OVLAN word argument (PCP bits). The PCF bit of this OVLAN word might be set to 1 meaning that the priority is to be seamlessly bridged through the intermediate nodes of the PLC cell. This priority information is summarized within the Priority field of any Burst Header and the Frame Priority field of every upstream data token.

8.2.2 Max Subcell Access Time

The Max_Subcell_Access_Time corresponds to the max duration for a flow to access the channel on the subcell level. It is a latency requirement for the scheduler to be configured within any master of a PLC cell.

Four Max_Subcell_Access_Time are defined at the master level. The minimum value of the Max_Subcell_Access_Time is used as the reference step from which the three other Max_Subcell_Access_Time are derived by multiplying by 2, 4 or 8 this reference step.

8.2.2.1 Configuration

The reference step shall be configured within any master of a PLC cell. This parameter is configured using the QOS_LATENCY_STEP parameter (min 20ms, max 400ms).

Mapping a Max_Subcell_Access_Time to a priority: Such a mapping shall be declared within each master by using the QOS_LATENCY.prio+1 parameter.

8.2.3 Resource reservation type

This parameter is related to the type of claimed guarantees provided in terms of node resources and time resources.

It can take three values:

- Type Best Effort: No guarantees on the node and time resources. No CAC accepted. Resources are granted on a Best Effort basis.

  Note: A maximum upstream and downstream bandwidth limitation applies to all the Best Effort flows of the same node (see Opera Specification: Part 2).

- Type VBR (Variable Bit-Rate): Only VBR type CAC requests accepted. VBR type CAC requests ask for node and time resources for which an average bandwidth can be guaranteed with a maximum variation around this average.

- Type CBR (Constant Bit-Rate): Only CBR type CAC requests accepted. CBR type CAC requests ask for individual guarantees in terms of node and time resources. The granted node and time resources are not shared. The claimed bandwidth is guaranteed.

Note: Any class which is associated to a VBR or CBR type corresponds to a connected service requiring initiation of a Connection Admission Control Procedure. By extension, CBR or VBR flows are called connected flows unlike Best Effort flows which are non connected flows.

8.2.3.1 Configuration

Mapping a Resource Reservation Type to a priority: Such a mapping shall be declared within each master by using the QOS_PRIORSV.prio+1 parameter.

8.2.4 Service reliability

This parameter defines if the ACK mode is enabled (see chapter 5.4) within the service class.

8.2.4.1 Configuration

Mapping Service Reliability to a priority: Such a mapping shall be declared within each master by using the QOS_PRIOACK.prio+1 parameter.

8.2.5 Example of a service class definition

Table 11 shows an example of possible relationship between service classes, priorities, Max_Subcell_Access_Time, Service Reliability and Resource Reservation Type.

Service Class	Priority	MaxSubcell Access Time	Service Reliability (ACK enabled)	Resource reservation
7	7	*240*	*No*	*Best Effort*
6	*6*	*30*	*No*	*CBR*
5	*5*	*60*	*Yes*	*VBR*
4	*4*	*120*	*Yes*	*VBR*
3	*3*	*120*	*Yes*	*VBR*
2	*2*	*120*	*No*	*VBR*
1	*1*	*240*	*No*	*Best Effort*
0	*0*	*240*	*No*	*Best Effort*

Table 11 Definition of Service Classes: example

The definition of service classes as described in table 1 could be used to implement QoS services for applications as described in Table 12.

Service Class	Application Examples
7	Management messages
6	*VoIP*
5	*Video, Games*
4	*Data Highprio*
3	*Data Highprio*
2	*Data Highprio*
1	*Data Lowprio*
0	*Data Lowprio*

Table 12 Mapping applications to service classes: example

8.2.6 Provisioning Service Class Definitions to slaves

As detailed in 9.1.6, nodes that wish to request a resource reservation through the Connection Admission Control procedure must make a reference to the requested service class. As a matter of fact, it is necessary for the slaves to have an idea of the service class definitions. The description of the available service classes will be obtained through the Autoconfiguration process.

8.3 Congestion Management

The mission of CM is to manage the accepted connections in congested networks (due to a sudden drop in the channel quality, for example). There are two possible policies that can be configured in the system:

- **Fair Congestion Management Policy (FCMP)**: all traffic flows are treated equally. Therefore, if the traffic scheduler cannot meet the bandwidth and latency requirements from users, it will decrease the performance of all active flows equally until new bandwidth and latency definitions can be supported.

- **Priority Congestion Management Policy (PCMP):** lower priority traffic performance is decreased in order to guarantee higher priority traffic SLA. Therefore, access to the channel is denied to lower service types first, until SLA of higher service types can be maintained.

8.3.1 Configuration

The Congestion Management can be configured using the Auto-configuration protocol. In this section, we enumerate the auto-configuration parameters that modify the behavior of the Congestion Management. For more information, see Part 2 of the specification (OPERA Standard System Specification) in the *System Parameters* Section (Section 4)..

QOS_BW_POLICY = [0|1] configures the CM in which the QoS manages the network in the event of congestion (0 configures the CM in FCMP mode; 1 configures the CM in PCMP mode).

8.4 Connection Admission Control

Initiation of a Connection Admission Control (CAC) procedure is required for any connected flow. The mission of the Connection Admission Control is to ensure that the PLC network can deliver the agreed services and guarantees to new flows while maintaining the requested performance of already accepted flows.

The Connection Admission Control functionality is embedded in master nodes. The CAC protocol might be used in the following two scenarios:

- Before the transmission of a connected flow, the CAC shall be used to reserve resources on the nodes in the system that collaborate in the communication of the new flow (from the node initiating the flow to the node terminating the flow). The CAC requests made to commit resources for a given Service Class may be new or updates from previous reservations.

- At the end of the transmission of a connected flow, resources may be released. Release of resources may be done explicitly by transmitting a CAC request message indicating no traffic requirement, or may be done when a timer, MAX_CACREQ_TO expires since the last reservation was done.

In the CAC protocol, the exchanged parameters to verify whether or not the PLC network can guarantee the required service shall include the required bandwidth, and the maximum latency demanded by the new node.

8.5 Token Fields

The required information to be exchanged included in the Data Token (see 4.4.2.1) and Burst Header (see 5.2) in order to perform all the scheduling tasks mentioned above is:

- Data Token Frame Priority Field (only in the upstream)

- Burst Header Priority

- Single Validity Field

- Received number of words Field

Furthermore, there are two more fields included in the Data Token going in the Upstream direction that give the scheduler an idea of the PLC topology so that the *Validity* field is correctly computed. These parameters are:

- Maximum Number of Hops

- Number of Downstream Nodes

9 LAYER MANAGEMENT

9.1 Control protocols

9.1.1 OPERACP (OPERA Communication Protocol) Description

The exchange of management messages between OPERA nodes needs a special and homogeneous structure in order to simplify implementation and clarify information exchanges between nodes. Such information will have as source and destination OPERA nodes, and will be used for internal management.

The structure of the packet that will be exchanged by two PLC nodes will follow the scheme presented in Figure 44:

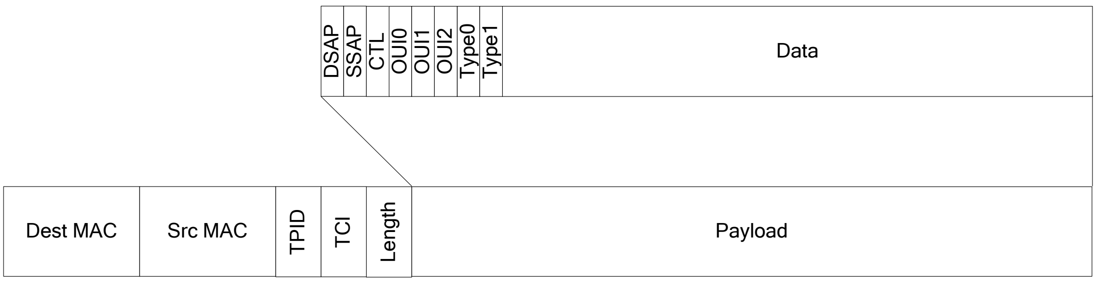

Figure 44 Management packet

The management information will be encapsulated using SNAP as depicted in Figure 44 into a regular Ethernet Frame. Using SNAP with an assigned OUI, it is possible to create specific protocols for OPERA nodes.

Encapsulated fields will be described in the following table:

Field	Length	Description
DSAP	1 byte	Destination Service Access Point
SSAP	1 byte	Source Service Access Point
CTL	1 byte	Control field
OUI0	1 byte	Organizationally Unique Identifier Byte 0
OUI1	1 byte	Organizationally Unique Identifier Byte 1

OUI2	1 byte	Organizationally Unique Identifier Byte 2
Type0	1 byte	Ethertype byte 0
Type1	1 byte	Ethertype byte 1

Table 13 Management packet field description

For the internal powerline protocols that OPERA stations will exchange, the described fields must have the following values:

Field	**Length**	**Value**
DSAP	1 byte	0xAA
SSAP	1 byte	0xAA
CTL	1 byte	0x3
OUI0	1 byte	0x00
OUI1	1 byte	0x13
OUI2	1 byte	0x9D
Type0	1 byte	See Table 15
Type1	1 byte	See Table 15

Table 14 Management packet field values

Table 15 includes the used types and subtypes for mandatory PLC protocols. Description of each one will be provided in the following sections. In general, with Type0 a Specific protocol is set and with Type1 a message type that makes sense into the protocol is announced. In the following table, messages of each PLC protocol are described, providing information about the content of each type field for each message.

Type0	**Protocol**	**Type1**	**Description**
0x01	Adaptive Bit-Loading Protocol	0x01	BPC
0x02	Port Solver Protocol	0x01	Access Message

		0x02	Announce message
0x03	Access Protocol	0x01	Petition Status
0x05	Connection Admission Control Protocol	0x01	CAC Request
		0x02	CAC Response
0x06	Encryption		See 10.2.4.1
0x07	Parametric Translation Table protocol		See OPERA system specification (Part II)
0x0a	Cluster Discovery Protocol	0x01	CDP message
0x0b	Automatic management of crosstalks	0x01	IDP
		0x02	BNDP
		0x03	BNDA

Table 15 Management packet protocol types

In some cases it is foreseen that the management information that has to be sent may not fit in one single packet. A mechanism is provided to split the information in several packets using sub-tables. The general format of those packets follows in Table 16:

Bytes	Fields	Length
0	Sub-table number	1 bytes
1	Number of sub-tables	1 bytes
2	Number of entries of the sub-table	1 byte
3 - 8	Sender MAC address	6 byte
9	Entry size	1 byte
10 - 521	Entries	512 bytes Maximum

Table 16 Packet format

The sub-table number shall be 1 for the first packet, and the sub-tables shall be sent in order. Entries can have variable length to allow for forward compatibility. The entry size is expressed in bytes. A node receiving entries with a size bigger than what it expects shall only take the values in the entries that it knows how to interpret and shall look for the next entry taking into account the entry size received in the packet.

The final structure, including information about Bridging and VLAN, and aligned in bytes shall be transmitted through the power line in the order presented in the following figure:

bits 7	6	5	4	3	2	1	0	Octet
LB[3]								1
LB[2]								2
LB[1]								3
LB[0]								4
PB[1]								5
PB[0]								6
LB[5]								7
LB[4]								8
PB[5]								9
PB[4]								10
PB[3]								11
PB[2]								12
OVLANID[7:0]								13
Reserved			OVLANV	OVLANID[11:8]				14
Reserved				PCP			PCF	15
Reserved								16
DA[3]								17
DA[2]								18
DA[1]								19
DA[0]								20
SA[1]								21
SA[0]								22
DA[5]								23
DA[4]								24
SA[5]								25
SA[4]								26
SA[3]								27
SA[2]								28
TCI[7:0]								29
TCI[15:8]								30
TPID[7:0]								31
TPID[15:8]								32
SSAP								33
DSAP								34
Lenght[1]								35
Length[0]								36
OUI[2]								37
OUI[1]								38
OUI[0]								39
CTL								40
Data[1]								41
Data[0]								42
TYPE1								43
TYPE0								44
Data from byte 2 to N								
PLW								

Figure 45 Management packet structure

When VLANs are active, all OPERACP packets must use the power-line reserved VLAN tag 1.

Other standard protocols like spanning tree, will follow the IEEE 802 standard.

9.1.2 Adaptive Bit-Loading Protocol (ABLP)

9.1.2.1 Introduction

The *Adaptive Bit-Loading Protocol* is in charge of the necessary interchange of the bit-loading tables between the different nodes of an established PLC network.

The bit-loading table information is used by a transmitter node to send data to other nodes in the network. The same configuration must be used by receiver nodes in order to understand the received data.

9.1.2.1.1 General Description

The ABLP sends bit-loading tables from the data receiver to the data transmitter. Once negotiation is correctly completed both nodes change to the new bit-loading table.

ABLP negotiation is unidirectional and needs direct (one hop) visibility between the transmitter and the receiver node.

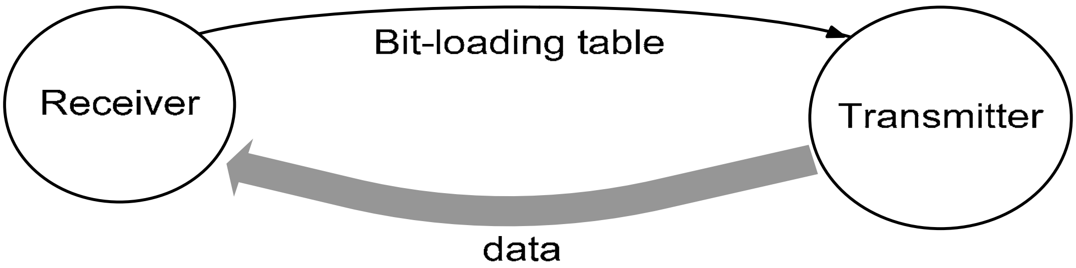

Figure 46 Bit-Loading

Bit-loading table transmission is done in the opposite direction of the traffic data that is going to use that bit-loading table (Figure 46). The receiver node sends the bit-loading table to the transmitter node. Then, this new bit-loading table will be used by the transmitter node to send data to the receiver node.

9.1.2.2 Protocol

The ABLP uses a “send packet, wait ack” approach.

ABLP packets are self-contained and they are never retransmitted. An acknowledgment mechanism is used by the receiver to know that the bit-loading table has arrived successfully to the other node and the transmitter node is going to use that new bit-loading table.

In case that acknowledgment (ACK) does not arrive, the receiver node will not use that bit-loading table. The protocol will be restarted with the transmission of a new ABLP packet. The loss of an ABLP packet will imply the use of the old bit-loading table. However, the loss of the acknowledgment will suppose that the transmitter node will use the new bit-loading table and the receiver node the old one.

Therefore, given that ACK replies are also susceptible to be lost, the protocol redundancy is increased sending each ACK several times. ABLP packet acknowledgments are sent as a field in the token symbol (see 4.4.2.1); therefore, ACK retransmission does not suppose any traffic increment. Figure 47 shows that mechanism.

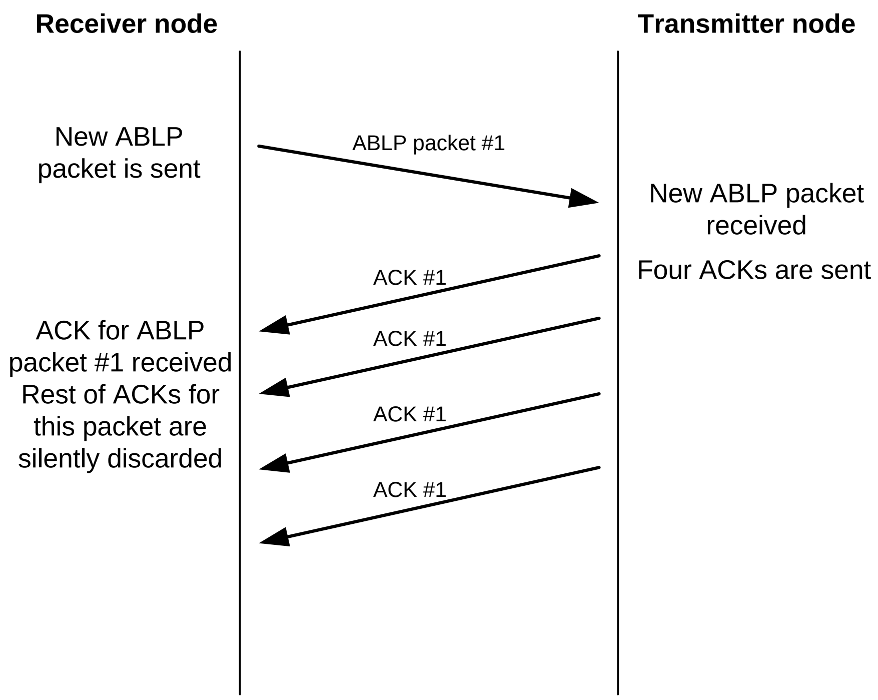

Figure 47 ABLP acknowledgement

The number of ACK retransmissions is a design variable. Redundant ACKs must be discarded (Figure 47).

First, the transmitter node must send the ACK. Then, it must change its bit-loading table in the following transmission frame. On the other hand, after receiving the ACK, the receiver node must change the bit-loading table before the next frame reception.

Periodically the receiver node will have a new bit-loading table to send. The time elapsed between consecutive ABLP packets is not defined in this specification.

9.1.2.3 ABLP packet format

ABLP packets are sent in HURTO mapping through port 127. That way, it will be possible to recover from any bit-loading table mismatch at the next ABLP exchange. The use of the unicast port would result in the impossibility to continue negotiation when mismatch has occurred. The packets use the OPERACP with Type0 0x01 and Type1 0x01. The packet format can be seen in Figure 48.

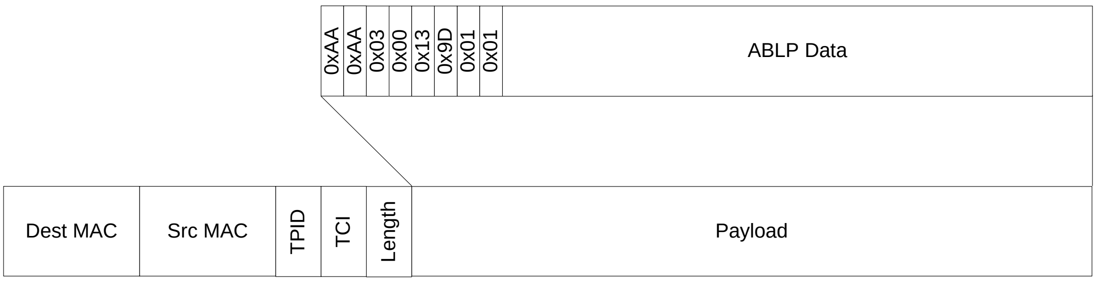

Figure 48 ABLP Packet

ABLP packets use the following multicast address as destination MAC (0x01139D000000). The transmitter must discard ABLP packets not addressed to it.

Source MAC is set to receiver node MAC. The ABLP data is shown in Figure 49.

Byte 0	0x00	
1	Remote Port	
2	bpcid	
3	Local Port	
4	reserved	H
5–6	bps	
7	Rx Att	
8–9	reserved	
10–15	Transmitter MAC	
16–19	crc	
20–21	reserved	
22–789	Bit-loading table	

Figure 49 ABLP Data

Description:

- Remote port: remote port with the ABLP transmitter

- Bpcid: ABLP packet sequence number (1 byte). The transmitter node does not need to receive ABLP packets in order; this value is only used for acknowledgment purposes. Packets with the same Bpcid will have the same ACK response.

- Local port: local port with the ABLP transmitter

- H bit (H means HURTO): used for notifying to the transmitter that it has to use HURTO mode. When this bit is set to 1 the bit-loading table is not included and the ABLP data size is reduced to 22 bytes.

- BPS value (2 bytes): (see Equation 17). This value is the BPS value calculated from the bit-loading table included in the packet. Format:

    - Data[5] = BPS & 0x00ff

    - Data[6] = (BPS & 0xff00) >> 8

- Rx Att: difference in 3dB steps between the maximum reception gain, and the gain used to receive the frame that generated this channel estimation.

- Transmitter MAC: network byte order (6 bytes). It contains the transmitter MAC address.

- CRC: this value is calculated considering as 0 the CRC field. It is the CRC-32 calculation from byte 0 to 21 or from byte 0 to 789, depending on the packet size. Polynomial:

$$c(x) = 1 + x + x^2 + x^4 + x^5 + x^7 + x^8 + x^{10} + x^{11} + x^{12} + x^{16} + x^{22} + x^{23} + x^{26} + x^{32}$$

  It is applied in little endian each 32-bit word (XOR bit-mask is 0xEDB88320). The 32-bit value is stored as:

    - Data[16] = CRC & 0x000000ff

    - Data[17] = (CRC & 0x0000ff00) >> 8

    - Data[18] = (CRC & 0x00ff0000) >> 16

    - Data[19] = (CRC & 0xff000000) >> 24

- Reserved: There are several reserved bits for future enhancements. These reserved bits must be set to 0.

- Bit-loading table: included at the end of the packet (768 bytes). Bit-loading table data is structured as 4 bits per subcarrier – 0, 2, 3, 4, 5, 6, 7, 8 ,9 or 10. Assuming that bpc[1] is the value assigned to the first subcarrier (lowest frequency) and bpc[1536] the value of the last subcarrier (highest frequency):

- Data[22] = (bpc[770] << 4) + bpc[769]
- Data[23] = (bpc[772] << 4) + bpc[771]
- ...
- Data[405] = (bpc[1536] << 4) + bpc[1535]
- Data[406] = (bpc[2] << 4) + bpc[1]
- Data[407] = (bpc[4] << 4) + bpc[3]
- ...
- Data[789] = (bpc[768] << 4) + bpc[767]

9.1.2.4 ABLP ACKs

The ABLP protocol is closely related to the MAC layer. ABLP packet acknowledgments are sent by the MAC layer in the data token symbol together with other MAC information (see 4.4.2.1).

9.1.3 Access Protocol

Access protocol defines a protocol for connecting nodes in point-multipoint power line networks within master-slave architecture.

In this protocol we will define the way to establish a connection from a slave to a master of the network.

Each master node may periodically send an access frame, as explained in 4.3.5, to the network, and wait for a response to add the responding slave to their own list and then start the AAA (Authorization – Authentication – Accounting) protocol to decide if the node is authorized to connect to the network via this master or not.

On the other side, when a slave receives an access token it decides to reply or not with an access reply frame, as explained in 4.3.6.

When a slave wants to send an access reply frame it must start a back-off period, as it is explained in 4.3.6. If no other access reply frame is detected during that period, the node may send an access reply frame.

Once a master receives an access reply frame from a slave it must send an access protocol packet accepting or rejecting the slave.

9.1.3.1 Access Protocol packet format

This packet shall be sent from the master to a slave that sent and access reply frame, to indicate if the slave is accepted or not. The packet is formatted as an OPERACP packet and the structure is shown in Figure 50.

The response contained in the packet can be of three different types:

- ACCEPT: The slave node has been accepted
- REJECT: The slave node has been rejected because the AAA process has resulted in rejection.
- FAILED: The slave node has been rejected because the AAA process has failed and the master does not have enough information to accept or reject the node.

The packet uses the OPERACP with Type0=0x03 and Type1=0x01.

In Figure 50 we can see the entire packet format:

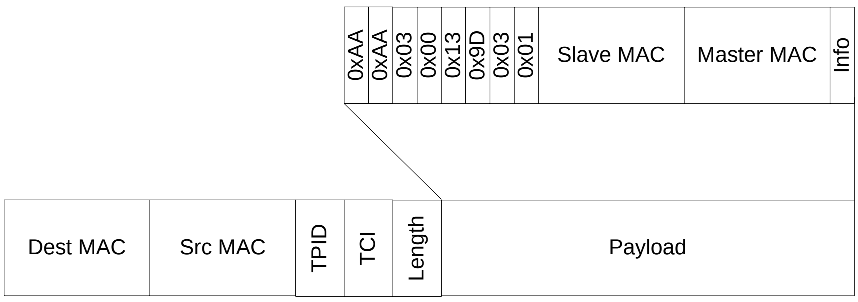

Figure 50 Access protocol packet

The packet is sent using the broadcast port (127). The value of the packet fields shall be:

- Dest MAC is set 0x0180C2000000.
- Src MAC is set to the master MAC address.
- Slave MAC is set to the slave MAC address
- Master MAC is set to the master MAC address
- The Info field is 1 byte and can have the following values:
    - REJECT: 0x00

- o ACCEPT: 0x01
- o FAILED: 0x02

9.1.3.2 Timing diagrams

In the following timeline diagrams we will show the sequence of messages that can be produced in access protocol between master and slave and the messages sent.

We will divide this section in different cases that can occur.

Case 01: Master accepts Slave

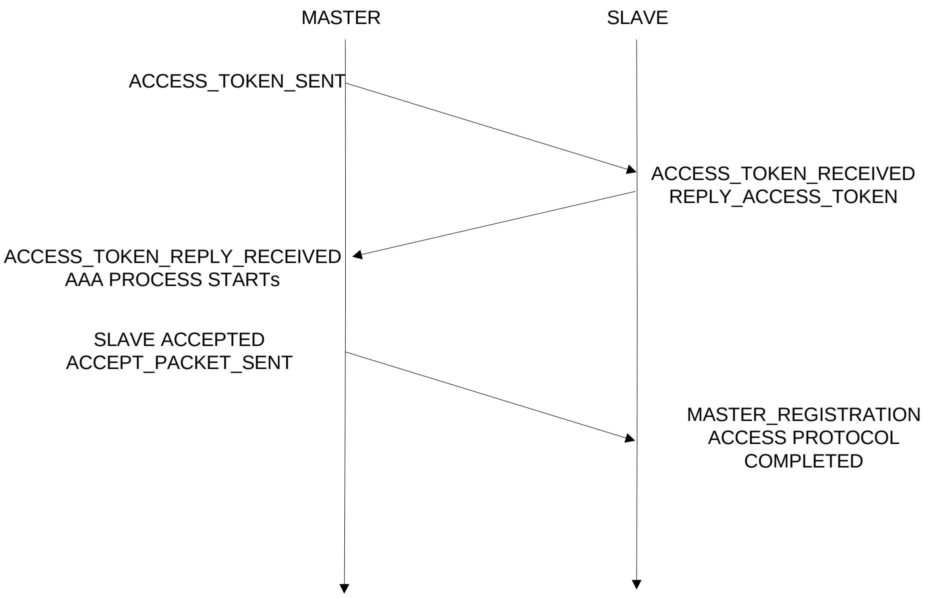

Figure 51 Access protocol: Master accepts Slave

Case 02: A Master rejects a Slave

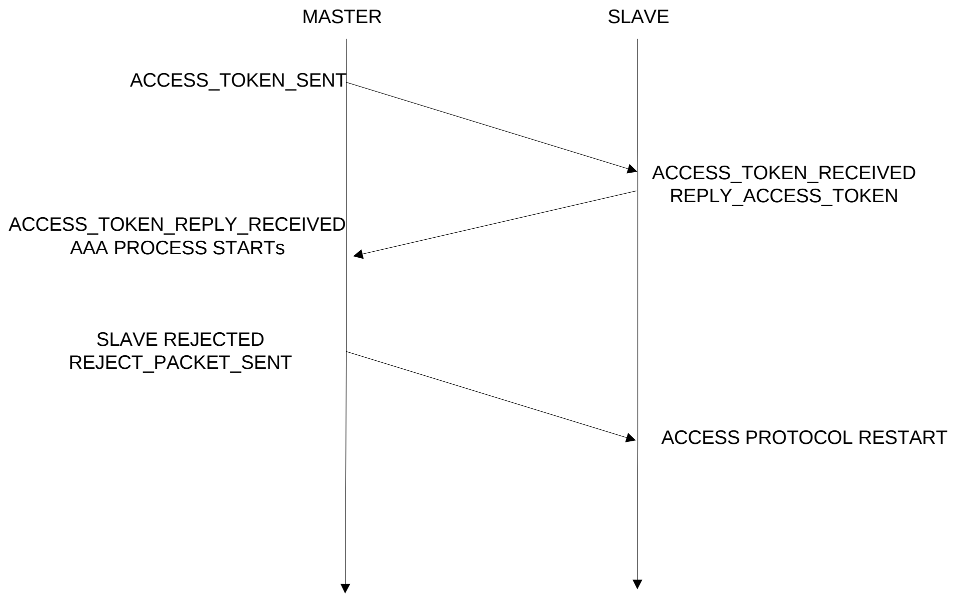

Figure 52 Access protocol: Master rejects Slave

Case 03 Master rejects Slave due to different reasons from not acceptation (protocol has failed)

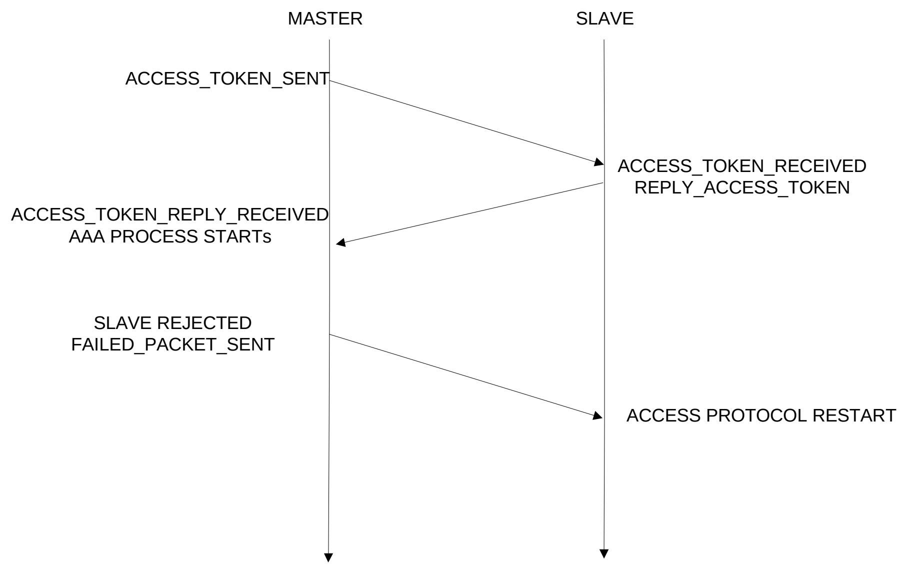

Figure 53 Access protocol: registration failed

Case 04: Slave response is lost

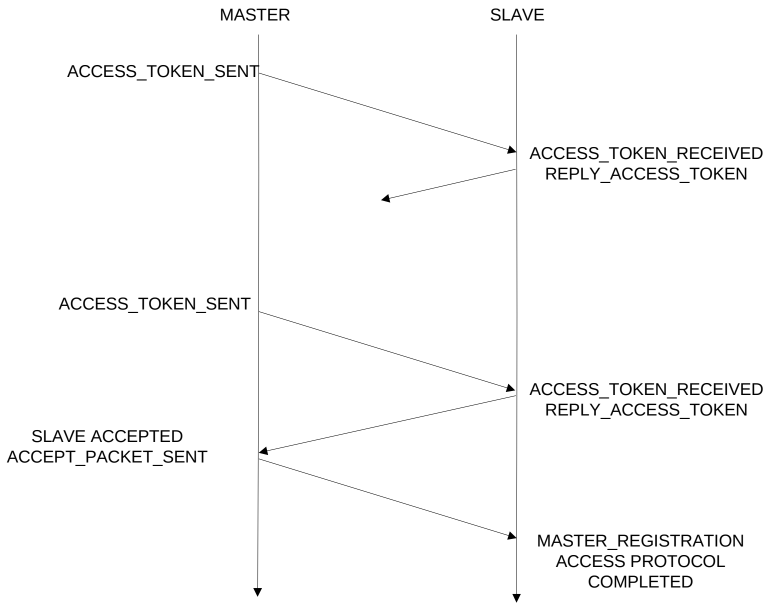

Figure 54 Access protocol: Slave response is lost

Case 05: Master acceptation packet is lost. Protocol starts again after expiring timers waiting for master reply.

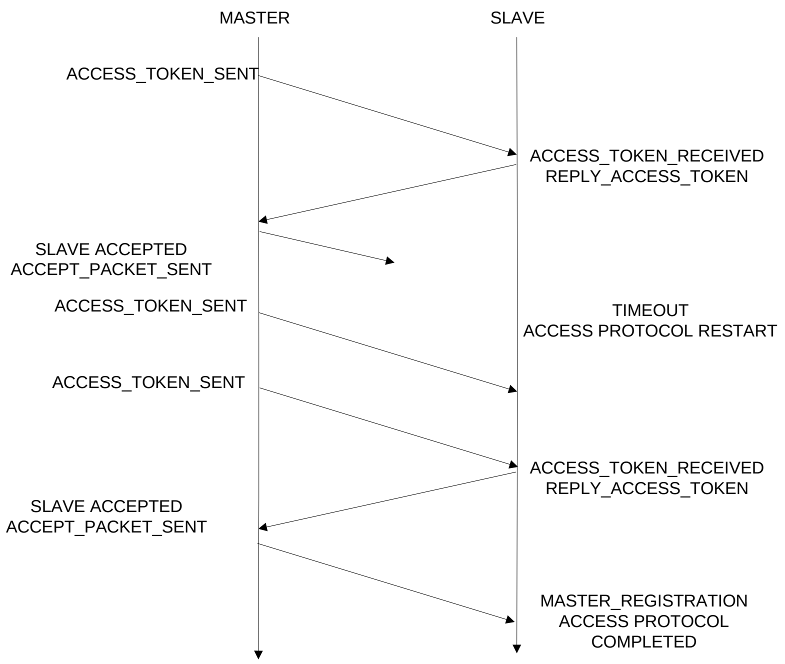

Figure 55 Access protocol: Master acceptation packet is lost

Case 06: Master accepts Slave 1. Slave 2 lost contention

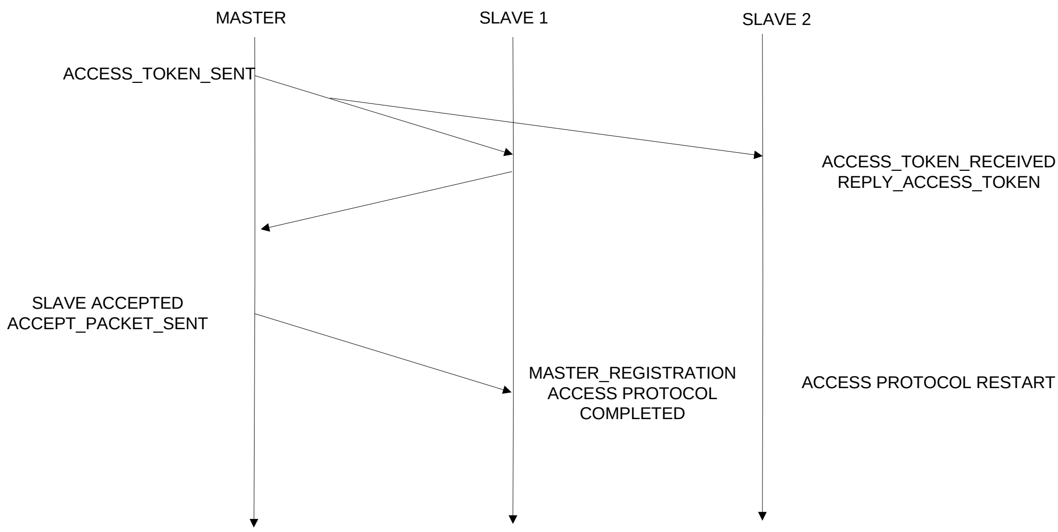

Figure 56 Access protocol: two slaves contend

Case 07: Slave decides not to log in a Master

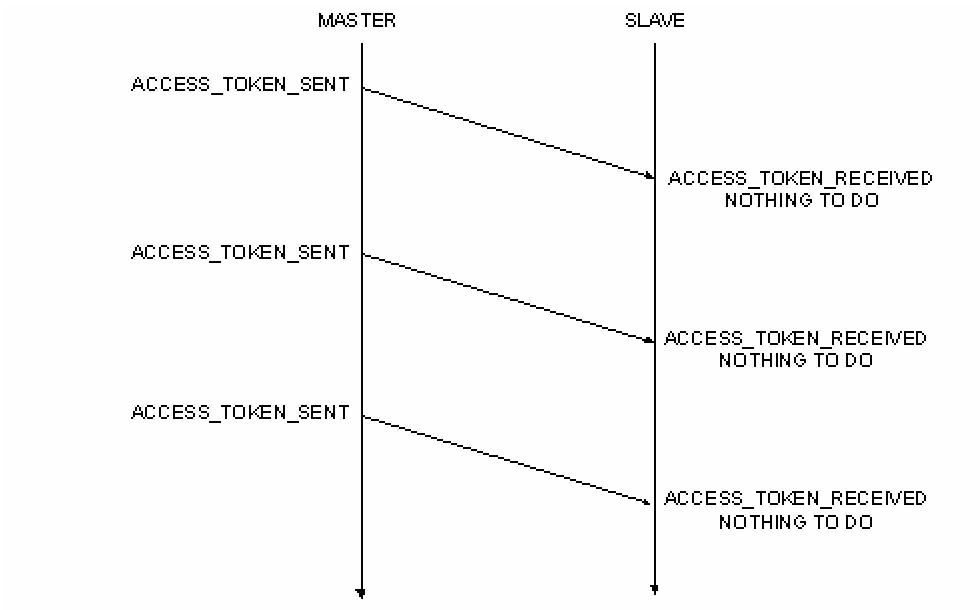

Figure 57 Access protocol: Slave decides not to log in a Master

9.1.4 Port Solver Protocol

To allow data transmission from node A to node B a connection must be created between them. This connection will allow sending data between A and B.

This connection is composed by two unidirectional links: one from A to B and another from B to A. These links are entries in the Port Solver Table (PST). That is, a connection between A and B is fully established when in A's PST there is a complete entry (with all its data) regarding B, and in B's PST there is a complete entry regarding A.

The objective of the protocol is to complete the remote port (RP) field of the PST entries.

9.1.4.1 Port Solver Table

The Token Announce is used to know the transmitter. In the Token Announce, there is the MAC address of the transmitter. For more information about the Token Announce see 4.4.2.

The way to identify the receiver at the transmitter side is with the Local Port. The Local Port is included in the Transmission Port field of the burst header.

At the receiver side, the information in the Transmission Port field of the burst header will be the Remote Port. The relation between Remote Port and MAC address of transmitter in order to identify that it is the intended receiver may be stored in a table (referred to as the Port Solver Table). To get more information about Burst Header see 5.2.

The transmitter can send bursts to up to 119 ports. The correct range of Local Port is the following:

- From 9 to 126 are possible Local Ports.

- Port 127 is the Broadcast Port. All receivers have to receive the burst with Transmission Port equal to 127.

The Port Solver Table has 118 entries and each entry has the following information:

- MAC address of all transmitters that are visible for this node. The MAC address of the transmitter can be read in the received Token Announce

- Remote Port (RP). The RP will be set to a default invalid value (0xFF) until the Port Solver Protocol takes place. The RP of transmitters that do not transmit data directly to this node (for instance, the RP of a CPE for another CPE) will keep the default value.

- Local Port (LP)

The Port Solver Table will have therefore completed entries (with a valid RP) and incomplete entries (with a RP equal to 0xFF). When receiving from a MAC address with a completed entry in the Port Solver Table, the receiver shall decode any burst with Transmission Port field in the burst header equal to:

- The Remote Port associated to this MAC address
- The Broadcast Port.

When Remote Port is using Mode ACK, the next sent ACK has ACK Port equal to Local Port.

Port Solver Table entries are deleted if no TA is received from the corresponding MAC address during 60 seconds (ageing).

The following example is useful to understand the meaning of each field:

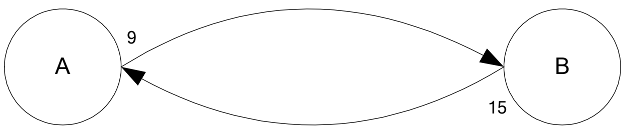

Figure 58 Ports example

To transmit a burst from A to B, the Transmission Port field is equal to 9.

To transmit a burst from B to A, the Transmission Port field is equal to 15.

In B the Port Solver Table has the following entry:

- MAC address equal to A MAC address
- Local Port equal to 15.
- Remote Port equal to 9.

In A the Port Solver Table has the following entry:

- MAC address equal to B MAC address
- Local Port equal to 9.
- Remote Port equal to 15.

9.1.4.2 Exchange process

The exchange process is only executed between the master and the corresponding slaves. The protocol is always started by the master, after completion of the access protocol. The slave node cannot send the port solver message because it needs the token to transmit any packet, and to recognize that the token is addressed to it, it must know its RP with the master; that is, it has to complete the entry in its PST correspondent to the MAC address of its master.

To complete all the fields of an entry correspondent to a given MAC address, the remote transmission port must be communicated by the node with that MAC address. A hand-shake protocol is used.

The node that wants to complete the entry sends a Port Solver Message (PSM) addressed to the MAC address of the PST entry. When a node receives a PSM, it answers with another PSM (Figure 59). The main contents of the PSM are the LP and the flag field. The flag field is set to zero by default. The flag field must be set to 1 when a PSM is received and the correspondent entry is already completed (Figure 61). A received PSM with this flag set to 1 will not require an answer.

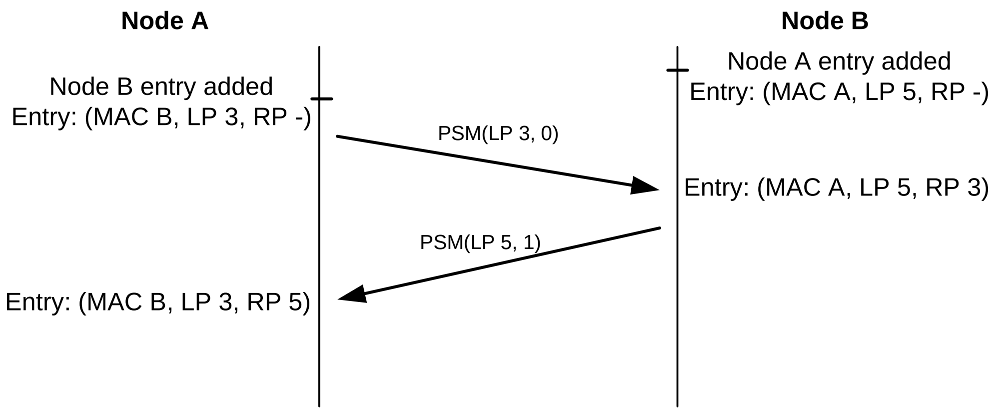

Figure 59 Port Solver Protocol Messages

If a response to PSM is not received, the entry will not be completed and a new PSM must be sent (Figure 60, Figure 61). There is a minimum time between PSM retransmissions. If the entry remains in the PST, the protocol must be followed until its success.

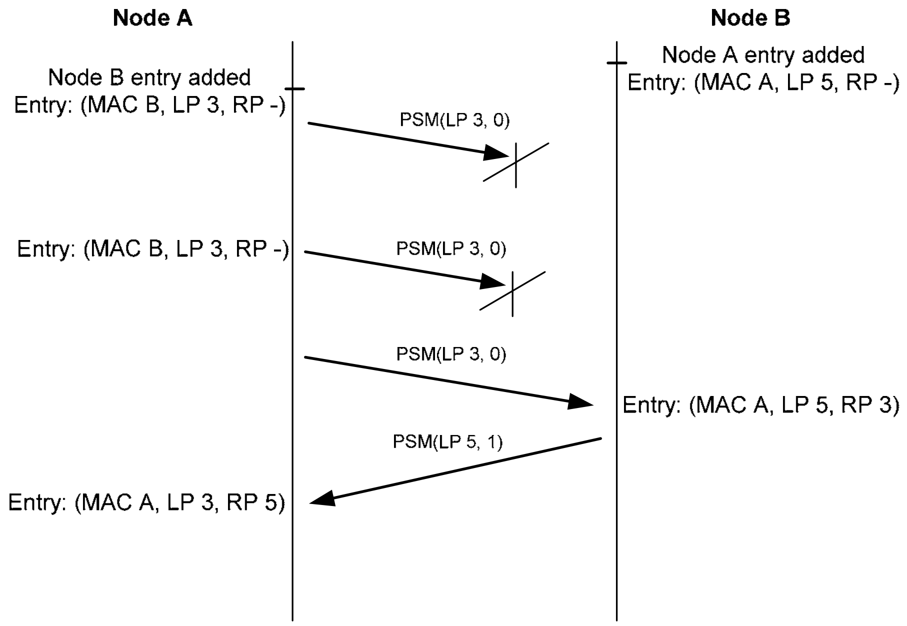

Figure 60 Port Solver Protocol Messages

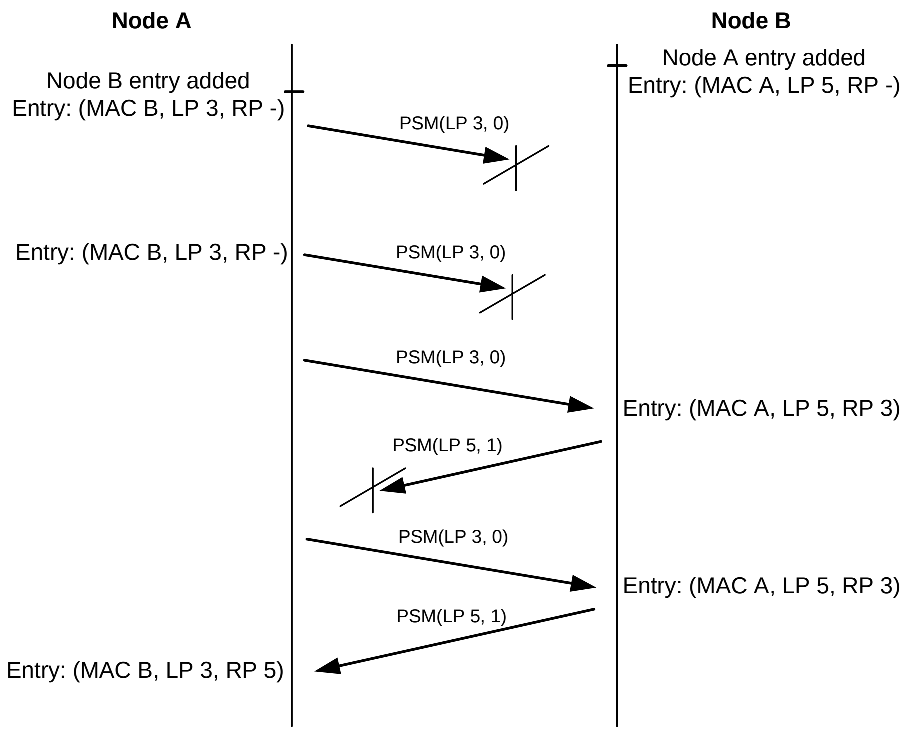

Figure 61 Port Solver Protocol Messages

Every time a PSM is received, the correspondent entry in the PST must be updated because the remote port can change.

Every time the LP is changed, the RP must be set to an invalid value and the protocol started.

9.1.4.2.1 Port Solver Packet format

The PSM is encapsulated using the OPERACP with Type0 0x02 and Type1 0x01 (Figure 62). The Port Solver packet will be sent through broadcast port 127.

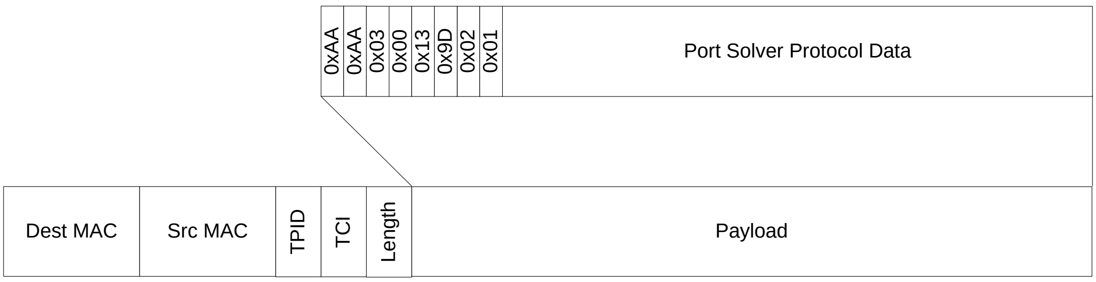

Figure 62 Port solver packet

Destination MAC must be the STP multicast address (0x0180C2000000) because PSMs are exchanged prior to the STP protocol. The packet shall be transmitted using the broadcast port (127). The PSM data fields are:

Bytes	PSM data fields	Length
0 – 5	Source MAC	6 bytes
6 – 11	Destination MAC	6 bytes
12	Assigned LP	1 byte
13	Info	1 byte

Table 17 Port solver packet fields

The format of the Info field is depicted in Figure 63:

7	6	5	4	3	2	1	0
Reserved							Flag

Figure 63 Info field format

9.1.4.3 Periodic publication of Port Solver information

Each Announce Time period (30 seconds) each node sends an Announce Message (AnM) publishing Port Solver information. This message contains a list with pairs of values (entry MAC address and assigned LP).

Upon the reception of an AnM, the PST shall be modified in the following cases:

- The sender MAC address is in a completed entry of the PST and the AnM LP value associated to the node MAC address is different from the PST RP value associated to the sender MAC address ➔ the RP shall be updated (see Figure 64)
- The sender MAC address is in the PST but the MAC address of the node is not in the AnM ➔ Set PST RP value associated to the sender MAC address to 0xFF (see Figure 65)
- The sender MAC address is in a incomplete entry of the PST, and the sender is either slave or master of the receiving node ➔ the RP shall be updated

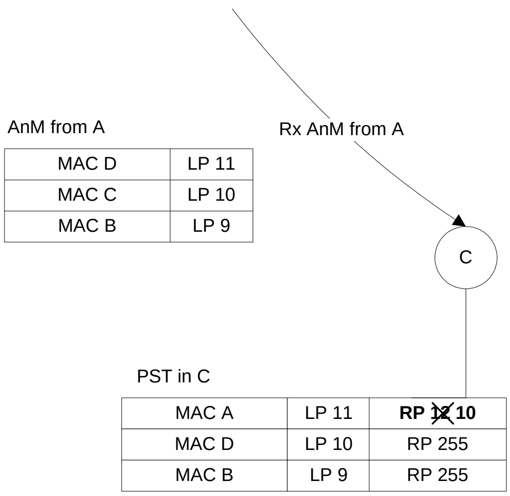

Figure 64 Announce Messages

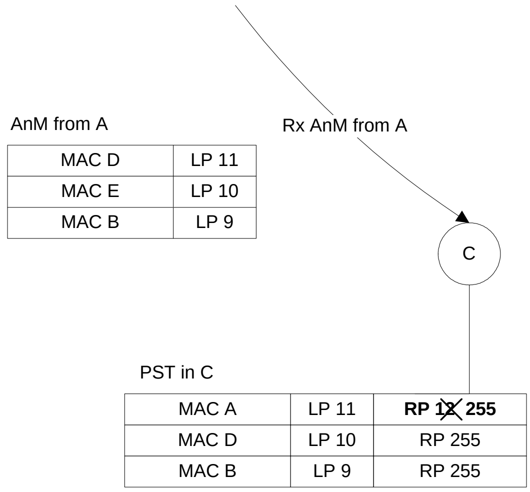

Figure 65 Announce Messages

This message does not require any acknowledgement.

9.1.4.3.1 Announce message format

The Announce Message is encapsulated using the OPERACP with Type0 0x02 and Type1 0x02 (Figure 66). The Announce Message packet will be sent through broadcast port 127.

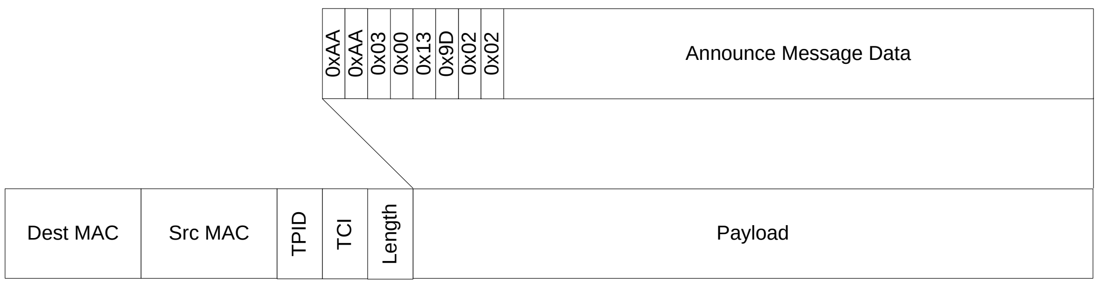

Figure 66 Announce packet format

Destination MAC must be the STP multicast address (0x0180C2000000).

The PST can be fragmented in several AnMs (identified by a sub-table number) if the entries do not fit in one.

The AnM data fields are:

Bytes	AnM data fields	Length
0	Sub-table number	1 bytes
1	Number of sub-tables	1 bytes
2	Number of entries of the sub-table	1 byte
3 - 8	Sender MAC address	6 byte
9	Entry size	1 byte
10 - 521	Entries	512 bytes Maximum

Table 18 Announce message fields

The Table data is a list of consecutive pairs in the form:

Entry fields	Length
MAC address	6 bytes
LP	1 byte

Table 19 Announce message entries

9.1.5 Cluster Discovery Protocol

The use of the Cluster Discovery Protocol (from here onwards, CDP) is limited to TDRs. Its use is not mandatory. The non-use of this protocol will result in the impossibility of its master to send non-returnable tokens.

A master must be able to determine if the nodes that are hanging from it may be divided in isolated clusters. In order to achieve this it is necessary for every node to transmit information in the upstream direction, in order to know which nodes are visible for each node. This is achieved through the Cluster Discovery Protocol.

CPEs are already sending the information regarding which nodes are visible with the periodic Announce Messages (see 9.1.4.3). They do not need to send any other information in the upstream.

However, a TDR X needs to send the following information to their masters:

1. List of nodes that are visible from X (this would be the same as the information in the periodic Announce Message sent by X);

2. List of nodes that belong to its PLC subtree: this includes CPEs and TDRs which are under the direct or indirect control of X;

3. List of nodes external to the PLC subtree of X that are visible from all the nodes belonging to its PLC subtree.

In order to transmit the information in points 2 and 3 the Cluster Discovery Protocol is used. This protocol simply consists in the periodic transmission from a TDR to its master of all this information. The information is recursive, because in order to form this packet, the TDR must have previously received the information from its slaves in order to include it.

The protocol will start in the first TDR (the one with only end-users hanging from it), once it has received at least one Periodic Announce Message from all of its slaves. The TDR will recollect this information, and send to its master a new packet. Every TDR in the chain will repeat this. Any TDR as well as the Head End might use the information received to determine that there are isolated clusters, and that it can perform spatial reuse by the use of the non-returnable token (see 4.4.2.6)

9.1.5.1 Cluster Discovery Protocol message format

The CDP Message is encapsulated using the OPERACP with Type0 0x0a and Type1 0x01 (Figure 66). The CDP packet will be sent through unicast port, as it is specifically addressed to the master node. The destination MAC will be the master node destination MAC.

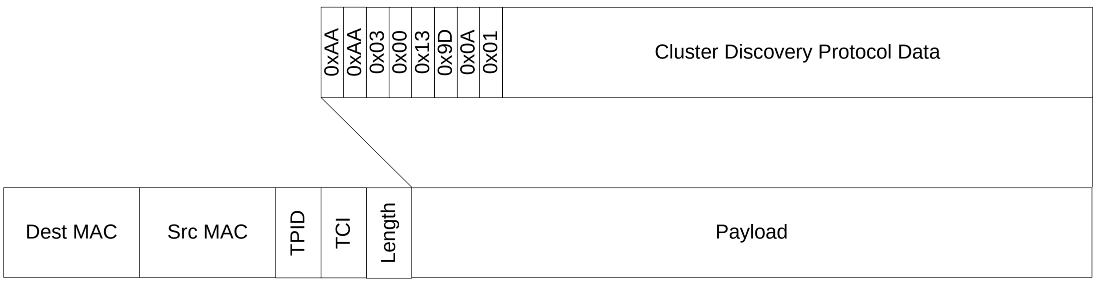

Figure 67 Cluster discovery packet format

The information can be fragmented in several packets (identified by a sub-table number) if the entries do not fit in one.

The CDP data fields are:

Bytes	CDP data fields	Length
0	Sub-table number	1 bytes
1	Number of sub-tables	1 bytes

2	Number of entries of the sub-table	1 byte
3 – 8	Sender MAC address	6 byte
9	Entry size	1 byte
10 - 521	Entries	512 bytes Maximum

Table 20 Cluster discovery packet fields

The Entries in the above table have the following format:

Entry fields	**Length**
MAC address	6 bytes
Dependency	1 byte

Table 21 Cluster discovery entries

The Dependency field will be set to 1 if the corresponding MAC belongs to the same PLC subtree. Otherwise, it will be set to 0.

9.1.6 Connection Admission Control Protocol (CAC)

The CAC Messages are sent whenever one or several new flow(s) register into the PLC network to reserve the type of traffic that should be assigned, and also when there are changes in the traffic requirements for any of the flows, so that new resources can be committed or released. If reservation parameters change during the duration of the current session, an update of the parameters is necessary.

There are two types of CAC messages:

- CAC_Req : Contains the resource reservation request
- CAC_Rsp ; Contains the response to a CAC_Req.

The CAC Messages are encapsulated using the OPERACP with Type0 0x05. CAC_Req message has the Type1 0x01. The CAC_Rsp message has the Type1 0x02.

Destination MAC address of CAC messages:

A Master CAC_Req message (see 9.1.6.1.2) is transmitted by the HE and it shall be addressed to the destination CPE to which the resource reservation is requested.

A Slave CAC_Req message is transmitted by a slave and shall be addressed to the master of the node transmitting the Slave CAC_Req.

A CAC_Rsp message is transmitted by a master as a deferred response to a Slave CAC_Req. In this case, it shall be addressed to the slave from which the Slave CAC_Req was originally received. A CAC_Rsp message can also be transmitted by a CPE as a reponse to a Master CAC_Req message. In this case, it shall be addressed to the HE.

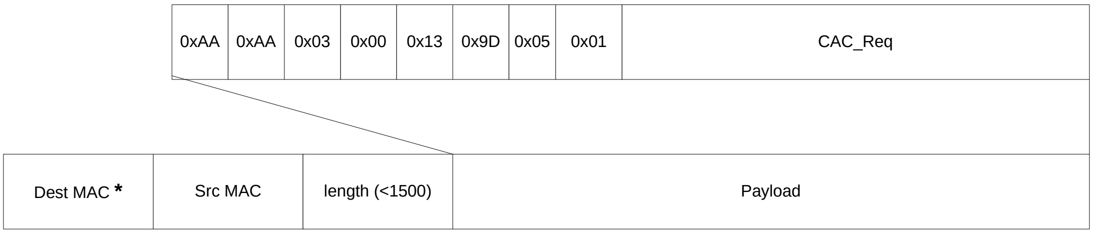

Figure 68 CAC_Req Message

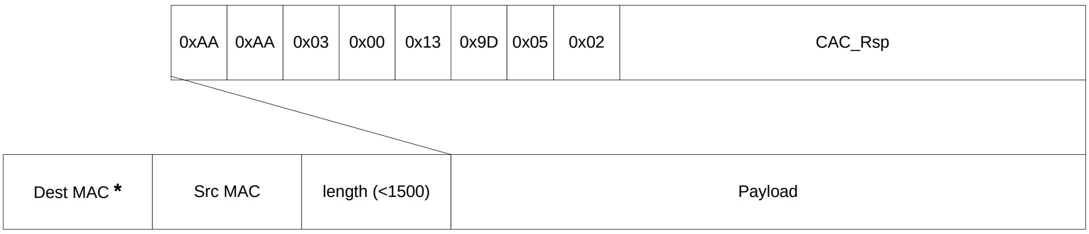

Figure 69 CAC_Rsp Message

9.1.6.1 CAC_Req

The CAC_Req message is transmitted to request VBR or CBR resources to be committed in order to satisfy the agreed quality parameters. It may originate from any node. There are two subtypes Slave CAC_Req and Master CAC_Req.

9.1.6.1.1 Slave CAC_Req

Slave CAC_Req messages are used to reserve the required upstream and downstream resources starting from a slave node. They are also used to maintain the requested resources (such messages are called Sustain Slave CAC_Req). Sustain Slave CAC_Req are a subtype of the Slave CAC_Req messages.

A Slave CAC_Req sequence is triggered by a CPE or upon reception of a Master CAC_Req issued by the HE (see 9.1.6.1.2). Before a CPE starts a flow requiring resources guarantees, it must trigger a CAC_Req sequence by sending a CAC_Req message to its master node. The master node then updates the service requests with the information received and relays a new CAC_Req message to its master node. This process is repeated until it reaches the HE.

Upon transmission of a Slave CAC_Req, the node shall freeze the corresponding requested resources and assign a Resource Identifier/Source MAC Address to these resources (same as the one included in the CAC_Req). It shall also start/reset two timers called MAX_CAC and MAX_CAC_REQ associated to this Resource Identifier/Source MAC Address. If the node does not receive any CAC_Rsp related to this Resource Identifier/Source MAC Address before MAX_CAC reaches MAX_CAC_TO (3 seconds), the node shall release the associated requested resources. If the node does not receive any Sustain Slave CAC_Req related to this Resource Identifier/Source MAC Address before MAX_CAC_REQ reaches MAX_CAC_REQ_TO (300 seconds), the node shall release the associated reserved resources.

Upon reception of a Slave CAC_Req by a master:

- If this Slave CAC_Req is a Sustain Slave CAC_Req,
    - If the Resource Request Description is coherent with the Resource Reservation in force for the requested Resource Identifier/Source MAC Address, the master shall maintain the reservation of the associated resources. Moreover, if this master is a repeater, it shall relay the Sustain Slave CAC_Req to its master keeping unchanged the Resource Identifier/Source MAC Address.
    - If there are mismatches between the Resource Request descriptions, the master shall discard the Slave CAC_Req.
- If this Slave CAC_Req is not a Sustain Slave CAC_Req,
    - if the master can grant the requested resources:
        - If the master is a repeater, it shall take into account this reservation request and then relay the request to its master, using the same Resource Identifier/Source MAC Address.
        - If the master is a HE, it shall take into account this reservation request and return an ACCEPT CAC_Rsp to the node from which the Slave CAC_Req has been received.
    - if the master can not grant the requested resources, it shall return a Reject CAC_Rsp to the node from which the Slave CAC_Req has been received.

9.1.6.1.2 Master CAC_Req

Master CAC_Req are used by a HE to communicate any given CPE to start the Slave Reservation CAC_Req with the indicated parameters. Such a request enables a HE to directly request, update or release a resource reservation. This message is issued by the HE and directly addressed to the destination CPE.

Upon reception of a Master CAC_Req by the CPE:

- If the CPE can take into account the request, the CPE shall acknowledge the request by sending an ACCEPT CAC_Rsp. It shall then trigger a Slave CAC_Req sequence using the same Source MAC Address, Resource Identifier and Resource Request Description as the ones included in the Master CAC_Req.
- If the CPE can not take into account the request, the CPE shall return a Reject CAC_Rsp to the HE.

Upon transmission of the Master CAC_Req, the HE shall start a MAX_CAC_REQ associated to this Resource Identifier. If the node does not receive any CAC_Rsp related to this Resource Identifier before MAX_CA_REQ reaches MAX_CAC_REQ_TO, the HE shall resend the Master CAC_Req.

Note: After transmission of a Master CAC_Req from the HE, the resource reservation as requested by the HE shall only be effective once the subsequent Slave CAC_Req is effectively received by the HE.

Note: CAC_Req messages may contain new resource requests or an update of a previous reservation. This update might eventually correspond to a partial release of the allocated resources (if the associated traffic requirements are decreased). It might also correspond to a total release of the allocated resources if the associated traffic requirements are set to 0.

The CAC_Req data fields are:

Bytes	CAC_Req data fields	Length
0 - 5	Source MAC address	6 bytes
6	Resource Identifier	2 bytes
7	Message Type	1 byte
8	Resource Request Description	Var

Table 22 CAC_Req Data Fields

Message Type	Value
Slave	0x00
Slave Sustain Notification	0x01
Master	0x02

Table 23 CAC_Req Message Type Field

2	Required DS Service Identifier	1 byte
3	Required US Service Identifier	1 byte
–-	DS traffic requirements	Var
--	US traffic requirements	Var

Table 24 Resource Request Description Field

The *Source MAC Address* is the MAC address of the HE if the CAC_Req is a Master CAC_Req (Message Type field is a HE). For Slave CAC_Req (Message Type field is a CPE), the *Source MAC address* is the MAC address of the HE if the request is originally triggered by a Master CAC_Req, else it is the MAC address of the originator CPE.

The *Resource Identifier* is a 16-bit number assigned to the CAC_Req. Combined with the *Source MAC address*, it uniquely identifies a US/DS Resource Reservation on the HE⇔ CPE path. An identifier value which has been used to send a CAC_Req message shall only be reused by a CPE or a HE if the corresponding Resource Reservation has been released by the node for more than MAX_CACREQ_TO.

The *Message Type* field indicates if this message is a Slave CAC_Req a Slave Sustain CAC_Req or a Master CAC_Req message.

The Required DS Service Identifier and Required US Service Identifier indicate the list of DS and/or US service classes that the node is requesting with reference to the Resource Identifier. The least significant bit represents service class 0 and the most significant bit represents service class 7. The bits related to Best Effort services shall not be set to 1 in this request.

The DS Traffic Requirements and/or US Traffic Requirements fields included in this request correspond to variable size fields. These fields carry the traffic requirements attached to each service class declared in the service identifier. First class declared in the DS Traffic

Specification (vice versa US) corresponds to the least significant bit set to 1 in the DS service identifier (vice versa US) and so on.

Two types of traffic requirements fields are defined in function of the two possible Resource Reservation Types (VBR or CBR). See Table 25 and Table 26.

CBR traffic requirements field	Length
CBR Bandwidth	2 bytes

Table 25 Traffic Requirements for CBR resource reservation

The Bandwidth field for CBR flows correspond to the maximum guaranteed bandwidth on the associated remote port for the given Service Class identified within the CAC_Req. It is in kBps (kilo bytes per second). Counting is performed over complete received packets when they are passed to the Convergence Layer.

VBR traffic requirements field	Length
Average Bandwidth	2 bytes
Maximum Bandwidth	2 bytes

Table 26 Traffic Requirements for VBR resource reservation

The Average Bandwidth field for VBR flows corresponds to a Guaranteed Average Bandwidth. It is in kBps (kilo bytes per second). Counting is performed over complete received packets when they are passed to the Convergence Layer.

The Maximum Bandwidth is the maximum sustainable bandwidth which would be delivered over a 100ms window if the average bandwidth recorded over the preceding 100ms window was equal to the Average Bandwidth field value. It is in kBps (kilo Bytes per second). Counting is performed over complete received packets when they are passed to the Convergence Layer.

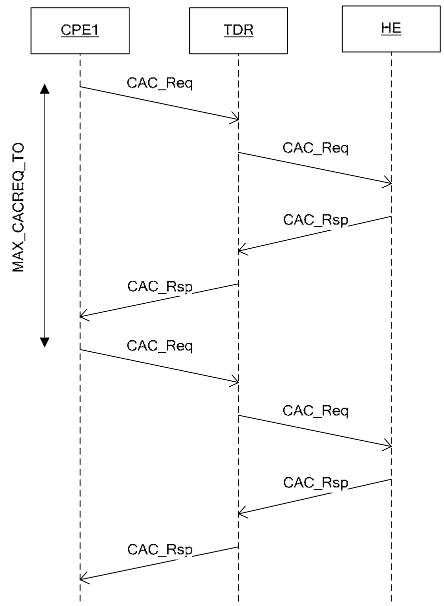

Figure 70 CAC Protocol Scenario 1

Figure 70 depicts a scenario where a new CPE sends a CAC_Req packet specifying its traffic requirement to its repeater. The repeater node relays the CAC_Req packet to the HE. Upon reception, the HE sends a CAC_Rsp packet with the response to the TD Repeater. This response is then forwarded to the CPE. After MAX_CACREQ_TO, the CPE shall update the resource request by sending a new CAC_Req message. If the HE does not receive the update, it assumes the connection has ended and resources related to this reservation will be released.

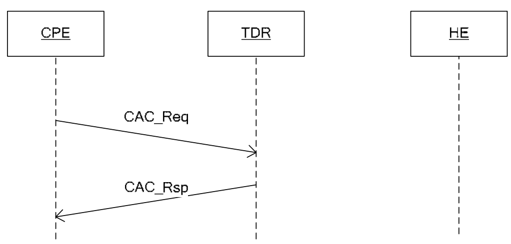

Figure 71 CAC Protocol Scenario 2

Figure 71 depicts a scenario where a new CPE sends a CAC_Req packet specifying its traffic requirements to its master node. If the master node cannot support the traffic requirements, it replies the CPE with a CAC_Rsp denying the service.

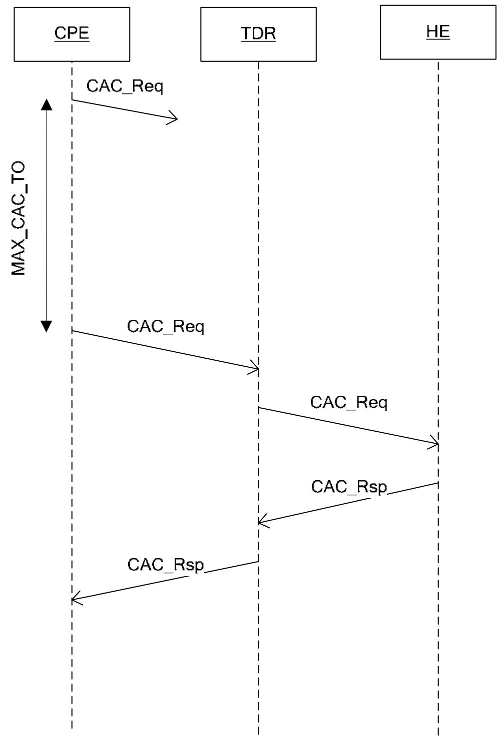

Figure 72 CAC Protocol Scenario 3

Figure 72 depicts a scenario where a CAC_Rsp packet does not reach the TDR, it is lost. After a MAX_CAC_TO time-out, the CAC_Rsp packet will be re-sent.

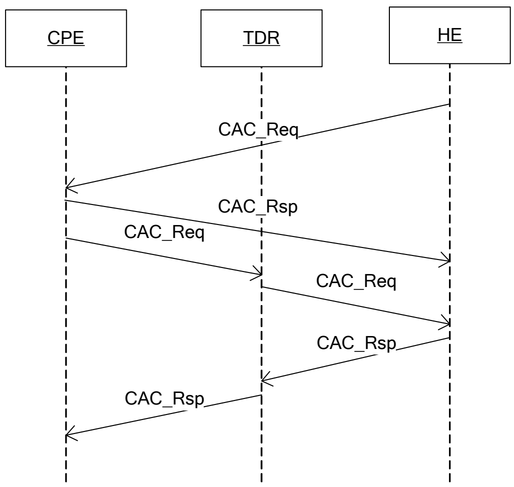

Figure 73 CAC Protocol Scenario 4

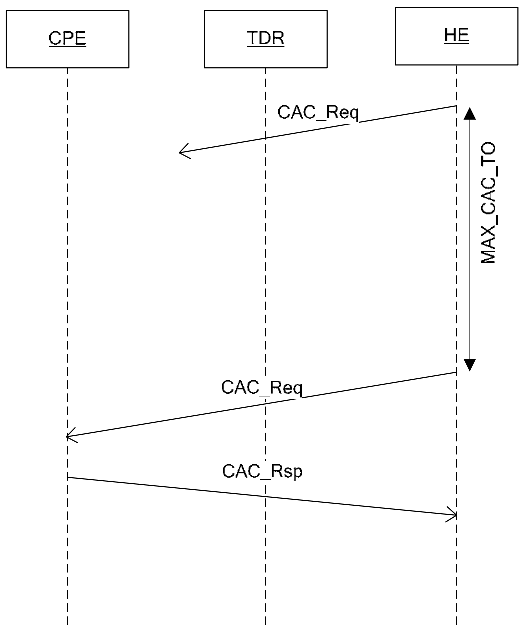

Figure 74 CAC Protocol Scenario 5

Figure 73 depicts a scenario where the HE starts the reservation protocol by sending a Master CAC_Req message to the CPE. Once it reaches the CPE, this node sends a CAC_Rsp acknowledging the reception of the Request and starts to reserve resources

Figure 74 depicts a scenario where the CAC_Req transmitted from the HE to the CPE is lost. After a timer of MAX_CAC_TO, if the HE has not received an CAC_Rsp acknowledging the reception of the CAC_Req, the HE will re-transmit the CAC_Req packet to the CPE.

9.1.6.2 CAC_Rsp

A CAC_Rsp message is transmitted:

- by a HE as a positive or negative response following the reception of a Slave CAC_Req;
- by a repeater as a negative response following the reception of a Slave CAC_Req;
- by a repeater as a relayed positive response following the reception of a CAC_Rsp issued by its master;
- by a CPE as a negative or positive response following the reception of a Master_CAC_Req.

The CAC_Rsp data fields are:

Bytes	CAC_Rsp data fields	Length
0 – 5	Source MAC address	6 bytes
6	Resource Identifier	2 bytes
7	CAC Response	1 byte

Table 27 CAC_Rsp Data Fields

The Resource Identifier field is equal to the Resource Identifier of the related CAC_Req.

The CAC Response field is coded as described in Table 28.

CAC Response	Value
Reject	0x00
Accept	0x01

Table 28 CAC Response

9.1.7 Automatic Management of Crosstalks between not Synchronized Systems

This section describes a mechanism to avoid crosstalk that may be implemented but is not mandatory.

Once the nodes realize that there is more than one master in the channel by means of the Master Id (see 4.4.1), it is necessary to take some measures to avoid such interferences.

The solution is to coordinate both PLC cells, and the simplest way to do it is subordinate one of the PLC cells. Next, the algorithm is explained:

9.1.7.1 MID hierarchy

When the crosstalk is detected, then the nodes must decide which PLC cell must subordinate.

The rule shall be "the lowest MID, the best", analyzing the MID as a number.

Then, one of the nodes of the worse PLC cell will register in the neighbor network using the Access Protocol, and it will act as master of its PLC cell, distributing the token.

If one node does not perform this protocol it does not copy the MID inherited from its master, instead it fixes the local MID to one of these values:

- 0x1FFFFF. Nodes that have not completed the Access Protocol.
- 0x130001. Nodes that have finished the Access Protocol.

9.1.7.2 Border node designation

The border node is the modem that will act as a bridge between the networks passing the token.

Once the interference is detected, the network with higher MID must decide which node within the PLC cell, will perform registration.

9.1.7.2.1 Designation Rules

The master will select the Border Node with the following criteria:

- Less hops, higher priority. This is, if the interference is detected by several nodes, the HE has preference over the rest of the nodes; next TDRs placed one hop away; next repeaters with two hops, etc.

- CPEs do not perform this protocol, only HEs and TDRs do. A node that detects only a CPE of a neighbor PLC cell does not apply for becoming border node.

9.1.7.2.2 Notification protocol

- To centralize the decision, the HE of the PLC cell decides which node will perform negotiation, interchanging OPERACP packets. Since the interference will not be detected by all the nodes simultaneously, a guard time may be kept before designating the border node in order to know all the information.

- If any node detects the interference (sniffing a better MID), it transmits an Interference Detection Packet (IDP), containing the own MAC, the neighbor MAC sniffed, the MID sniffed, the local number of hops and the identity of the node detected (HE or TDR). When the packet is received by a node, it has to check if is coming from one of its slaves, then the information is copied and retransmitted to its master, then the packet will arrive finally to the HE of the PLC cell. To avoid flooding the network, the IDPs have to be sent with a minimum period of 10 seconds between them, except if the node detects a better node to register to, this is new information that has to be transmitted immediately (for instance a node reported that was detecting a TDR but afterwards detects a HE, or another MID better than the transmitted one).

- Among all the IDPs received, the HE decides which node must become border node sending a Border Node Designation Packet (BNDP), containing the MAC of the Border Node. This is a broadcast packet that has to be relayed by all the nodes of the network because the destination might be several hops away from the master. When the packet is received by a node coming from its master and it is not the designated border node, it has to built a new BNDP to notify its slaves (only for TDR with at least one slave).

- The designated border node must send a Border Node Designation Acknowledge (BNDA), to confirm that the information has been received, and it is transmitted towards the master in the same way than the IDP. The BNDA must be received by the HE in a specific period of time (5 seconds). If the time expires a new BNDA to the same node is delivered. After 3 consecutive timeouts not receiving the BNDA, a new node (if exists) must be designated as Border Node.

9.1.7.2.3 Crosstalk Management Frames Format

9.1.7.2.3.1 IDP

The IDP Message is encapsulated using the OPERACP with Type0 0x0b and Type1 0x01 (Figure 75).

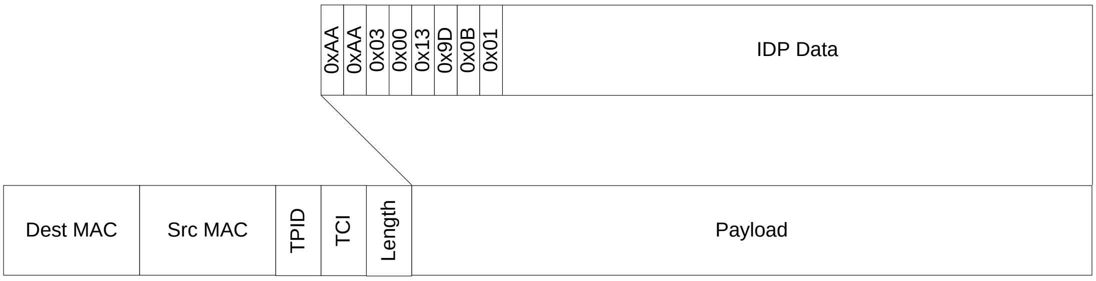

Figure 75 IDP Packet

The packet is sent using the broadcast port (127). The value of the packet fields shall be:

- Dest MAC is set 0x0180C2000000.
- Src MAC is set to the local MAC address.

The IDP data fields are:

Bytes	IDP data fields	Length
0-5	Sender MAC address	6 bytes
6-11	Sniffed MAC address	6 bytes
12-14	MID sniffed	3 bytes
15	Local Number of hops	1 byte
16	Type of Node sniffed	1 byte

Table 29 Interference detection packet fields

The field Type of Node contains the same information allocated in the Token Announce (although announce the detection of a slave is forbidden by the protocol).

0	HE
1	CPE
2	TDR

Table 30 Type of node in IDP

9.1.7.2.3.2 BNDP

The BNDP Message is encapsulated using the OPERACP with Type0 0x0b and Type1 0x02 (Figure 76).

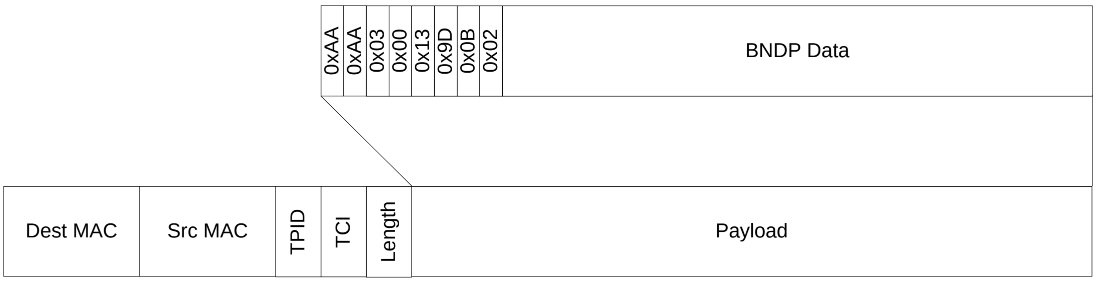

Figure 76 BNDP Packet

The packet is sent using the broadcast port (127). The value of the packet fields shall be:

- Dest MAC is set 0x0180C2000000.
- Src MAC is set to the local MAC address.

The BNDP data fields are:

Bytes	BNDP data fields	Length
0-5	Border Node MAC address	6 bytes

Table 31 BNDP packet fields

9.1.7.2.3.3 BNDA

The BNDA Message is encapsulated using the OPERACP with type 0x0b and subtype 0x03 (Figure 77).

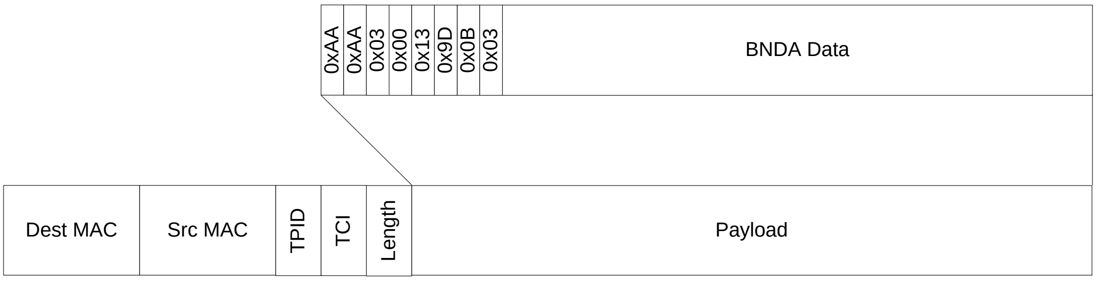

Figure 77 BNDA Packet

The packet is sent using the broadcast port (127). The value of the packet fields shall be:

- Dest MAC is set 0x0180C2000000.
- Src MAC is set to the local MAC address.

The BNDA data fields are:

Bytes	BNDA data fields	Length
0-5	Border Node MAC address	6 bytes

Table 32 BNDA packet fields

9.1.7.3 Border node registration

Once the border node is designated by the HE (can be itself), it starts its attempt for registration in the neighbor PLC cell as it is described in the Access Protocol.

9.1.7.3.1 Master selection

The border node will select the master to register based in the following criteria:

- First HE (of the neighbor PLC cell)
- Second repeaters.

9.1.7.3.2 Successful registration

If the modem succeeds, as it has changed PLC cell it copies the MID of its new PLC cell, in this way any node knows that the node is token dependant from the neighbor network.

A node performing the registration in the neighbor PLC cell will have a MID different and it will be different from 0x1FFFFF and 0x130001. Under these conditions the node must be accepted avoiding any authentication process, also the master node knows that is not a node from its network and must avoid transmit any data through the PLC port.

9.1.7.3.3 Failing registration

The maximum time specified for the registration of the border node is 120 seconds

When the timer expires, the HE has to select a new border node to complete negotiation. The HE is the node that monitories the registration timer, in this way no new packets are needed, even if the border node designated is switched off the HE restarts the protocol by itself.

The formerly designated border node has to activate the registration timer like the master and when the timer expires it assumes that it is not border node anymore until a new BNDP is received.

9.1.7.4 Intermittent transmission

To perform successfully the negotiation among PLC cells it is necessary to get free interference periods, because the transmissions within the own PLC cell will ruin the registration.

A HE can decide by itself to stop the transmissions within its network avoiding new token generation, but any other node cannot because its master will regenerate the token.

When the HE receives a BNDA it starts a period so called “INTERMITTENT TRANSMISSION” [IT].

During the status of Border Node a modem can generate as many BNDAs as needed to complete the negotiation.

The default silence time 2 sec, followed by a normal transmission period of 1 sec. It will have 5 silence periods. So the overall IT will last 15 seconds.

9.2 Spanning Tree

In a network, a spanning tree is a path or collection of paths that represent connections between nodes. To be called a spanning tree, the tree must cover every possible path in a network. A minimal spanning tree is one that covers all possible paths, does so with as few segments as possible, and makes sure there are no loops (closed paths) in the network.

The IEEE 802.1 recommendations provide an algorithm for finding a spanning tree in any network.

STP packets will be transmitted through the corresponding unicast ports.

9.2.1 IEEE 802.1D (Spanning Tree Protocol)

The 802.1D recommendation describes the Spanning Tree algorithm and protocol. In this document, the calculation of the tree elements and the messages exchanged between bridges for this purpose are exposed. This specification is fully compliant with this standard.

The implementation of IEEE 802.1D is a must for network compatibility.

For specific information on message types and their fields and the algorithm to handle them, refer to *IEEE Std 802.1* and *IEEE Std 802.1D (Spanning Tree)*.

Ports can be Ethernet or PLC. Every new PLC connection is added to the bridge as a new port. Ports can be deleted when the PLC connection is lost for some physical reason. Some ports, such as Ethernet ports, are always present. Each port carries out the STP.

9.2.1.1 Default timers and field values

The IEEE recommendation is relatively flexible in some aspects, such as default timer values and other message fields, so that they can be tuned for different network sizes and Medium Access technologies. The values adopted by this specification for these parameters are:

- **hello:** time between each BPDU that is sent on a port.
  - Default: 2 seconds
  - Range: [1, 10] seconds
- **forward delay:** time spent in the listening and learning states.
  - Default: 15 seconds
  - Range: [4, 30] seconds
- **max age:** maximum length of time a bridge port saves its configuration.
  - Default: 20 seconds
  - Range: [6, 40] seconds

- **Priority:** The greater the number, the less priority.

MV Master	0x9010
MV Repeater	0x9020
LV Master	0x9030
LV Repeater	0x9040
LV Slave	0x9050

Table 33 STP priority

9.2.2 IEEE 802.1w (Rapid STP)

Rapid Spanning Tree Protocol (RSTP; IEEE 802.1w) is an amendment to 802.1D. This version offers a significant reduction in the time taken to reconfigure the active topology of the Bridged LAN in the face of changes to the physical topology or its configuration parameters. The strategy to reach this performance is based on a handshake mechanism between bridges that enables a fast transition to forwarding, thus bypassing the listening and learning states and timers.

The two versions of the algorithm and protocol are capable of interoperating within the same Bridged LAN; hence, it is not necessary for implementations to support both versions of the Spanning Tree algorithm and protocol.

In view of the improved performance offered, it is recommended that the Rapid Spanning Tree algorithm and Protocol be supported in preference to the original version.

For specific information on message types and their format, the migration protocol and the RSTP algorithm, refer to *IEEE Std 802.1D, 1998 Edition; Clause 17 (Rapid Spanning Tree Algorithm and Protocol)*, also found as *802.1w*.

9.2.2.1 Default timers and field values

- **migration delay (mdelay):** indicates that the protocol migration is ongoing. It is used to avoid re-entry. If any BPDU of the other type is received while mdelay is running, it is ignored in terms of protocol migration. mdelay is set to 3 seconds.

9.2.2.2 Point-to-point ports and edge ports

In order to achieve fast convergence on a port, the protocol relies upon two new variables: edge ports and link type.

All PLC ports can be assumed to be point-to-point ports. Ethernet ports can be considered edge ports, unless another bridge is connected to the same segment. Anyway, since the standard establishes that the first BPDU coming into an edge port removes its edge port status, it can be assumed at startup that all ports are candidate to directly transition to forwarding:

- Ethernet ports start in forwarding state (edge ports).
- PLC ports, even though they start in blocking state, are ready for handshake and rapid transition to forwarding (point-to-point).

9.2.3 STP and VLANs

If VLANs are active, STP packets must use the management VLAN inside the power-line network.

9.2.3.1 Common STP

Ethernet interfaces must support the possibility of being configured to remove/add the VLAN tag to the outgoing/incoming STP packets. That is, the VLAN tag of the STP packets going out the PLC network is removed. On the other hand, VLAN tag must be added to the incoming STP packets, so as to make sure that STP packets inside the power-line network use the management VLAN.

10 SECURITY

10.1 Low level encryption

10.1.1 Overview

The cryptographic processing is performed over the data symbols of each burst independently.

In the burst header a bit called Crypt that is located at the byte 15, 6th bit. Burst header fields are described in section 5.2.

The encryption block will process this unit as follows. First, a random key denoted K_e is generated. Such key will be used to encrypt by means of DES algorithm the corresponding data that follows the burst header. In order to transmit such key K_e to the receiver, this random key (K_e) is encrypted using the 3DES algorithm with a private key called K_{tdes} shared by transmitter and receiver. The encrypted K_e by means of 3DES algorithm is introduced as part of the burst header.

The described procedure is repeated for each burst header, and the random key K_e generated for each of them. The private key K_{tdes} is different for each of the transmission ports. In addition to that, each communication direction from and to a node has an independent K_{tdes}.

This port oriented encryption algorithm allows an easy implementation of standards as 801.1x.

10.1.2 Detailed encryption process

First step of processing is the random generation of K_e. This key will have 64 bits length, of which 56 bits are the key and 8 bits are filling bits, in the positions 0, 8, 16, 24, 32, 40, 48 and 56, as shown in the following figure:

Ke[63:57]	Fill	Ke[55:49]	Fill	Ke[47:41]	Fill	Ke[39:33]	Fill	Ke[31:25]	Fill	Ke [23:17]	Fill	Ke[15:9]	Fill	Ke[7:1]	Fill

Figure 78 DES Key structure

Mathematically:

$$K_e[8*i] = Fillbits \quad i = 0,1,2\ldots7$$

Equation 36

Those filling bits are jointly generated with the key, and are also random generated. Then, this key K_e will be encrypted using the Triple DES algorithm as described in the FIPS PUB 46-3 document and the method described as ECB mode in FIPS PUB 81, using as key K_{tdes} the associated to the destination port to whom the encrypted data will be sent. This key K_{tdes} has 192 bits length, of which 168 will be the key that will be used as it following the FIPS PUB 46-3 and 24 bits are the filling bits, in the positions 0, 8, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176 and 184, as shown in the following figure:

Ktdes[191:185]	Fill	Ktdes[183:177]	Fill	Ktdes[175:169]	Fill	Ktdes[167:161]	Fill	Ktdes[159:153]	Fill	Ktdes [151:145]	Fill	Ktdes[143:137]	Fill	Ktdes[135:129]	Fill
Ktdes[127:121]	Fill	Ktdes[119:113]	Fill	Ktdes[111:105]	Fill	Ktdes[103:97]	Fill	Ktdes[95:89]	Fill	Ktdes [87:81]	Fill	Ktdes[79:73]	Fill	Ktdes[71:65]	Fill
Ktdes[63:57]	Fill	Ktdes[55:49]	Fill	Ktdes[47:41]	Fill	Ktdes[39:33]	Fill	Ktdes[31:25]	Fill	Ktdes [23:17]	Fill	Ktdes[15:9]	Fill	Ktdes[7:1]	Fill

Figure 79 Triple DES Key structure

Those filling bits are also randomly generated jointly with the key.

Mathematically:

$$K_{tdes}[8*i] = Fillbits \quad i = 0,1,2\ldots23$$

Equation 37

The encryption process is depicted in the following figure:

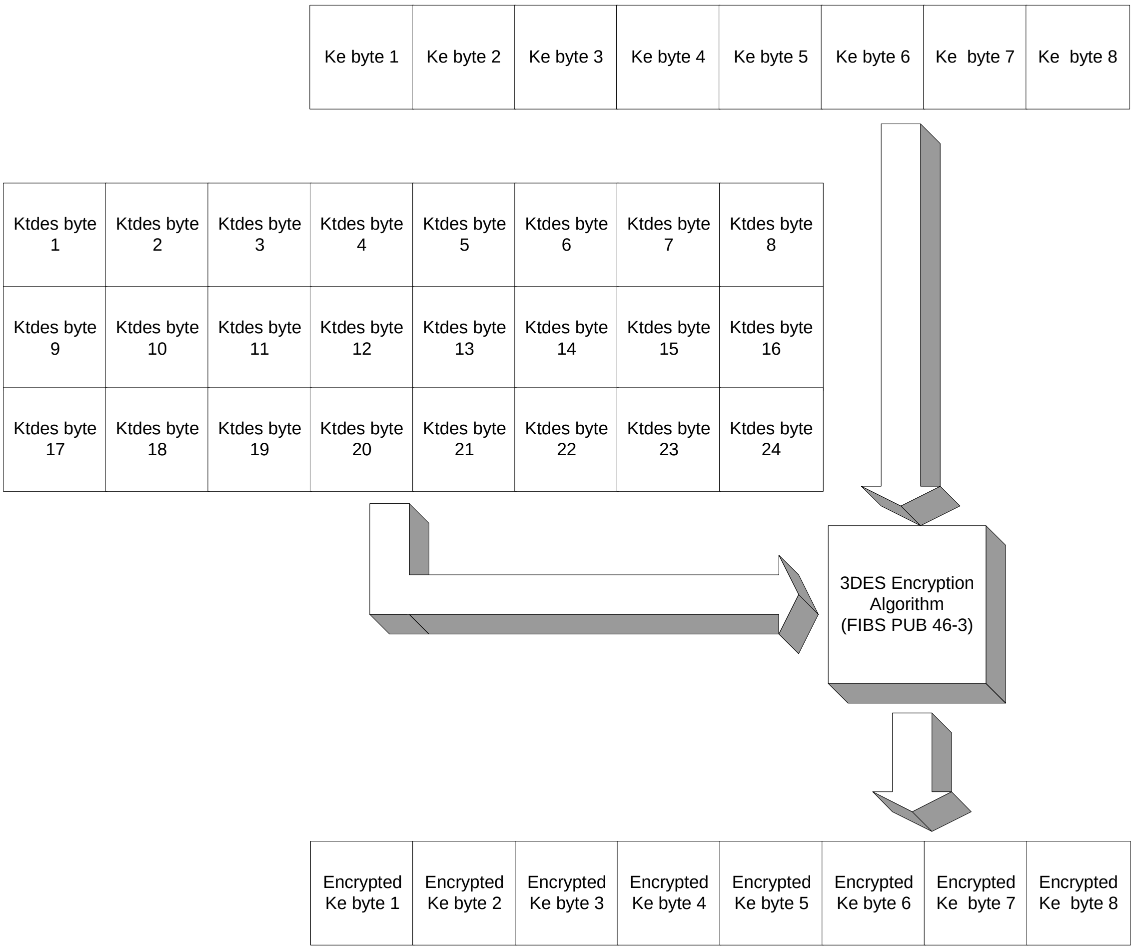

Figure 80 DES Key encryption process

From the K_{tdes} structure, only the bits identified as not filling bits will be effectively used to perform the encryption of the whole K_e structure, including the filling bits.

Once the K_e encryption process has finished, the resulting encrypted K_e will be included and transmitted in the burst header. In the transmission side, K_e is used to encrypt, by means of DES algorithm, the data associated to such burst header. The DES algorithm is described in FIPS PUB 46.3 and the method used can be read at FIPS PUB 81, appendix B, and is know as electronic codebook (ECB) mode.

The data will be divided into 64 bits portions, taken first the first to transmit byte that will be the most significant byte in the 64 bits and then enough bytes to complete the necessary 64 bit portion. Then, those 64 bits will be encrypted and will replace in the same position as original the structure that will be transmitted. In the case than the data to encrypt is not divisible by 64, the remaining bits will not be encrypted, and transmitted “as is” through the transmission line.

From the complete 64 bits K_e structure, only the 56 non-filling bits are used to perform the DES encryption algorithm.

The following figure illustrates the data encryption process:

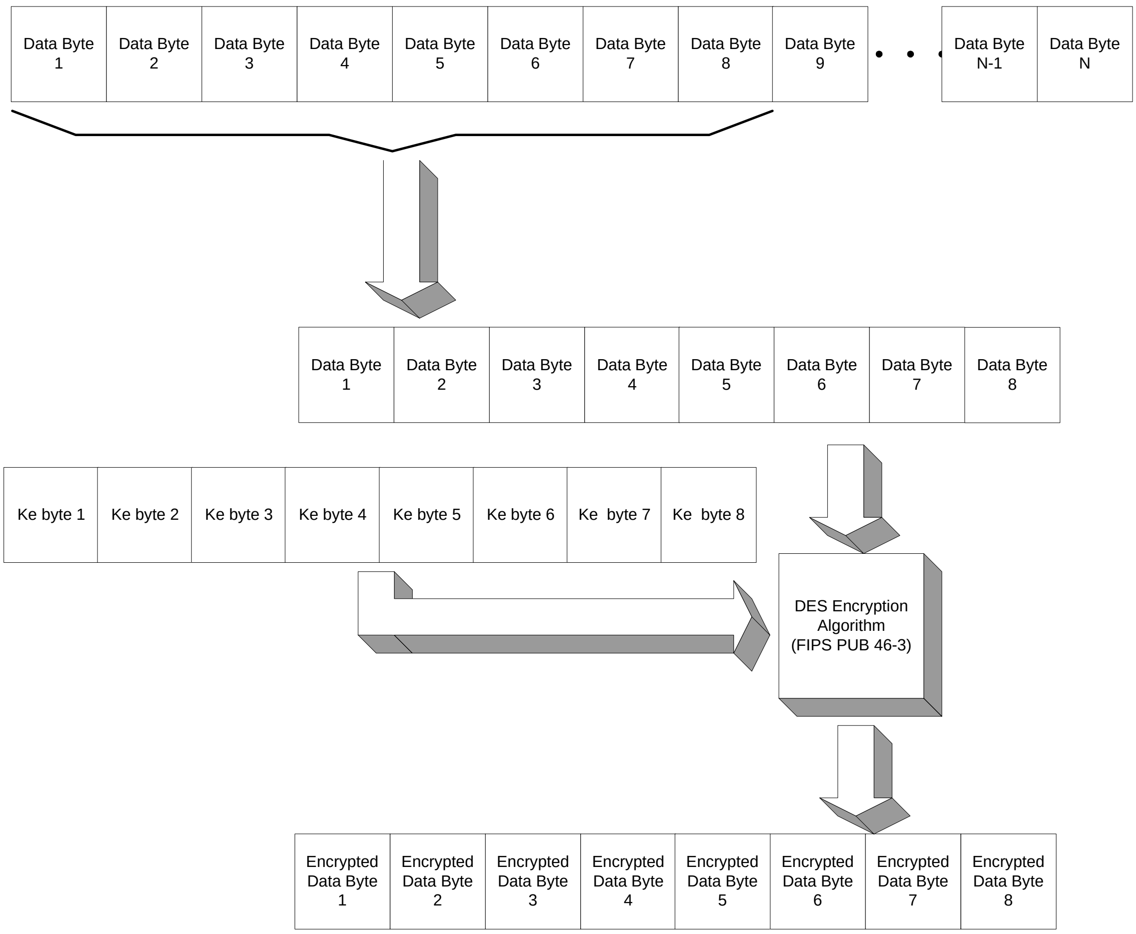

Figure 81 Data encryption process

On the other hand, the encrypted K_e key, will fill the bytes in the burst header denoted by byte 21, 22, 25, 26, 27, 28, 31 and 32, in this order, as described in section 5.2.

10.2 Key Encryption Negotiation Protocol

10.2.1 Introduction

The OPERA Key Exchange Protocol is based on the ANSI X9.42 specification.

10.2.2 Overview

Diffie-Hellman key agreement requires that both the sender and recipient of a message have key pairs. By combining one's private key and the other party's public key, both parties can compute the same shared secret number. This number can then be converted into cryptographic keying material. That keying material becomes into the content-encryption key (CEK) which is in turn used to encrypt the message data.

The final stage of the key agreement process is to compute a shared secret number, called ZZ. When the same originator and recipient public/private key pairs are used, the same ZZ value will result. The ZZ value is then converted into a shared symmetric cryptographic key. When the originator employs a static private/public key pair, the introduction of a public random value ensures that the resulting symmetric key will be different for each key agreement.

10.2.3 Generation of ZZ

X9.42 defines that the shared secret ZZ is generated as follows:

$$ZZ = g^{xb*xa} \bmod(p)$$

Equation 38

Note that the individual parties actually perform the computations:

$$ZZ = yb^{xa} \bmod(p) = ya^{xb} \bmod(p)$$

Equation 39

ya is party a's public key defined as:

$$ya = g^{xa} \bmod(p)$$

Equation 40

yb is party b's public key defined as:

$$yb = g^{xb} \bmod(p)$$

Equation 41

xa is party a's private key

xb is party b's private key

p is a large prime

q is a large prime

g is defined by the following formula:

$$g = h^{\frac{p-1}{q}} \bmod(p)$$

Equation 42

where h is any integer with 1 < h < p-1 such that:

$$h^{\frac{p-1}{q}} \bmod(p) > 1$$

Equation 43

This relationship between g, p and q means that g has order q mod(p), then:

$$g^{q} \bmod(p) = 1 \quad if \quad g \neq 1$$

Equation 44

That means that the relationship between p and q must be:

$$p = q * j + 1$$

Equation 45

Where j is a large integer number.

The standard X9.42 requires that the group parameters be of the form p=jq + 1 where q is a large prime of length m and j>=2. Algorithms to generate such primes of this form can be found in FIPS PUB 186-2 and X9.42.

X9.42 requires that the private key x be in the interval [2, (q - 2)]. x should be randomly generated in this interval. y is then computed by calculating g^x mod p. To comply with this memo, m must be >=160 bits in length, (consequently, q must be at least 160 bits long). Larger m may be advisable. p must be a minimum of 512 bits long.

10.2.4 Protocol description

10.2.4.1 Protocol Data Unit description

Protocol encapsulation will be done by means of of the OPERACP described at 9.1.1. The Type0 field will be set to 0x06 for this protocol messages. Each message is distinguished from others by means of the Type1 field. Each message has a particular structure and length.

The structure of each message used in the protocol is described in the following table.

Type1	Message name	Content	Length (Bytes)
0x00	TX_PRIMES	"g"	1
		"p"	Variable (up to 128)
0x01	ACK_TX_PRIMES	None	0
0x02	TX_PUBKEY	"y" ("ya" or "yb")	Variable (up to 128)
0x03	ACK_TX_PUBKEY	None	0

Table 34 Message types

10.2.4.2 Protocol message exchange

Each node has a p and g prime numbers pre-calculated. Private key is generated by means of these numbers, after first message exchange that will contain the common p and g prime numbers. The protocol will be initiated by the first node, which realizes the presence of another node that requires an encrypted communication.

In the following table, the sequence is described:

Node a	Node b
Send message TX_PRIMES, containing g and p from node a. Calculate using selected p and g the public key ya	
	Send message ACK_TX_PRIMES. Calculate using g and p received from node a the public key yb.
Send message TX_PUBKEY containing ya	

	Send message TX_PUBKEY containing yb. Calculate common key ZZ.
Send message ACK_TX_PUBKEY. Calculate common key ZZ. Start using encryption.	
	Start using encryption.

Table 35 Protocol message exchange

11 ACCESS / IN-HOUSE COEXISTENCE

OPERA access systems and inhouse systems might interfere with each other when transmitting over overlapping frequency bands. To enable coexistence of both systems, this specification refers to the first version of the Coexistence Specification proposed by the Universal Powerline Association (UPA). This protocol provides mechanisms to coexist in time and/or frequency.

The UPA Coexistence Specification is attached in Annex B. It is likely that UPA will revise this version in the near future. OPERA is willing to maintain coordination between both efforts and will update the OPERA technology specification accordingly.

ANNEX A PERFORMANCE (INFORMATIVE)

The total bitrate transmitted by a PLC node is influenced by several factors. Considering PHY level, the maximum theoretical rates depend on the OFDM Symbol Type (see 3.7.1) and spectral efficiency, which is bound by the bit-loading of each subcarrier (see 3.6).

Approximate numerical values will be given for maximum bit rates, taking into account these factors:

- OFDM symbol rate and cyclic prefix
- Truncated 4D-TCM
- Loss rate due to RS coding
- CRC overhead (Delimiters)

And specifically not considering these:

- Zero filling (Mapping modes, see 3.4)
- Special coding of last Reed-Solomon codewords (Data payload, see 3.3.2)
- Overhead of reference signals (see 3.8) when fitting OFDM symbols inside PPDU (see 3.2).

For simplicity purposes, numerical values will only be given for the special cases in which all 1536 subcarriers have the same bit-loading. This makes calculations easier, but it should not be difficult to find corresponding numerical values for any possible bit-loading table.

A.1 Data stream

Data payload (see 3.1) is adaptively or HURTO modulated.

The cyclic prefix that each Symbol Type uses (see 3.7.1) implies that actual data is only transmitted during the IDFT interval. The percentage of maximum PHY rate that is lost equals the percentage of cyclic prefix interval over total symbol interval (which is the sum of cyclic prefix and IDFT interval). The IDFT interval is fixed to 2048 samples (see Table 7) for every Symbol Type.

Truncated 4D-TCM operates on 768 pairs of subcarriers (see 3.5.1).

For adaptive mapping, each pair of subcarriers gets a different coding according to its bit-loading: if we call z to the sum of the bit-loadings of both subcarriers, the coding rate for that pair of subcarriers is $\frac{z-1}{z}$, except for the case in which the bit-loadings for both subcarriers is zero (in this case there is no coding because no message in being sent on that pair of subcarriers). As bit-loadings can take the values 0, 2, 3, 4, 5, 6, 7, 8, 9 or 10, coding rates will range from 1/2 to 19/20 for each pair of subcarriers.

For HURTO mapping, each subcarrier will get the same coding, with a bit-loading of 2, so the coding rate for each pair of subcarriers is 3/4. Additionally, HURTO mapping replicates eight times the message, which means a further coding rate of 1/8.

Next table shows coded maximum data rates (in Mbps) for OFDM symbols with constant bit-loading for all subcarriers. Numbers shown take into account cyclic prefix and 4D-TCM.

	Bit-loading for all 1536 subcarriers (adaptive mapping)									HURTO
	10	9	8	7	6	5	4	3	2	
Type I	204,94	183,37	161,80	140,22	118,65	97,08	75,51	53,93	32,36	4,04
Type II	150,82	134,95	119,07	103,19	87,32	71,44	55,57	39,69	23,81	2,98
Type III	84,01	75,16	66,32	57,48	48,64	39,79	30,95	22,11	13,26	1,66

Table 36 Coded maximum bitrates for Data payload (constant bit-loading)

Furthermore, RS coding (see 3.3.2) implies an additional rate loss which depends on the code rate. For each of the four RS data modes (0 being the least and 3 the most robust one), adaptive mapping can produce a total of 28 different codes depending on the FEC payload, each one with a different data rate as given in Table 37.

	FEC payload (bytes) / Code rate																											
RS mode 0	**244**	**240**	**236**	**232**	**228**	**224**	**220**	**216**	**212**	**208**	**204**	**200**	**196**	**192**	**188**	**184**	**180**	**176**	**172**	**168**	**164**	**160**	**156**	**152**	**148**	**144**	**140**	**136**
	0,968	0,968	0,967	0,967	0,966	0,966	0,965	0,964	0,964	0,963	0,962	0,962	0,961	0,960	0,959	0,958	0,957	0,957	0,956	0,955	0,953	0,952	0,951	0,950	0,949	0,947	0,946	0,944
RS mode 1	**240**	**236**	**232**	**228**	**224**	**220**	**216**	**212**	**208**	**204**	**200**	**196**	**192**	**188**	**184**	**180**	**176**	**172**	**168**	**164**	**160**	**156**	**152**	**148**	**144**	**140**	**136**	**132**
	0,952	0,952	0,951	0,950	0,949	0,948	0,947	0,946	0,945	0,944	0,943	0,942	0,941	0,940	0,939	0,938	0,936	0,935	0,933	0,932	0,930	0,929	0,927	0,925	0,923	0,921	0,919	0,917
RS mode 2	**236**	**232**	**228**	**224**	**220**	**216**	**212**	**208**	**204**	**200**	**196**	**192**	**188**	**184**	**180**	**176**	**172**	**168**	**164**	**160**	**156**	**152**	**148**	**144**	**140**	**136**	**132**	**128**
	0,937	0,935	0,934	0,933	0,932	0,931	0,930	0,929	0,927	0,926	0,925	0,923	0,922	0,920	0,918	0,917	0,915	0,913	0,911	0,909	0,907	0,905	0,902	0,900	0,897	0,895	0,892	0,889
RS mode 3	**232**	**228**	**224**	**220**	**216**	**212**	**208**	**204**	**200**	**196**	**192**	**188**	**184**	**180**	**176**	**172**	**168**	**164**	**160**	**156**	**152**	**148**	**144**	**140**	**136**	**132**	**128**	**124**
	0,921	0,919	0,918	0,917	0,915	0,914	0,912	0,911	0,909	0,907	0,906	0,904	0,902	0,900	0,898	0,896	0,894	0,891	0,889	0,886	0,884	0,881	0,878	0,875	0,872	0,868	0,865	0,861

Table 37 RS coding rates for Data payload (adaptive)

HURTO mapping can use all the RS codes above **plus** codes with decreasing FEC payload until code rate is 0.5.

The uncoded maximum bitrates for Data payload are the result of multiplying coded maximum bitrates by RS coding rate.

A.2 Delimiters

Delimiters (see 3.1) are HURTO modulated for robust OFDM transmission.

Control symbols have modulation and trellis coding schemes similar to HURTO data symbols, so the maximum coded data rate is the same.

Delimiters use a specific RS code with rate 2/3 (see 3.3.1.2). Additionally, a CRC-CCITT is applied which implies a rate loss of 176/192.

So in the same conditions as assumed for Data payload, the Delimiters data rate is given in Table 38

	Uncoded bitrate (Mbps)
Type I	2,47
Type II	1,82
Type III	1,01

Table 38 Coded maximum bitrates for Delimiters

ANNEX B UPA COEXISTENCE SPECIFICATION (NORMATIVE)

B.1 Introduction

This Annex includes the first version of the Coexistence Specification of the Universal Powerline Association, UPA - www.upaplc.org.

B.1.1 Scope

The scope of this document is to describe mechanisms to address coexistence issues between one Access system and several In-home systems and between several In-home systems without any Access system.

The powerline systems that do not make use of frequencies between 2 MHz and 30 MHz are not in the scope of this document.

The following example shows two systems A and B each with several devices that share a part of the same electrical installation.

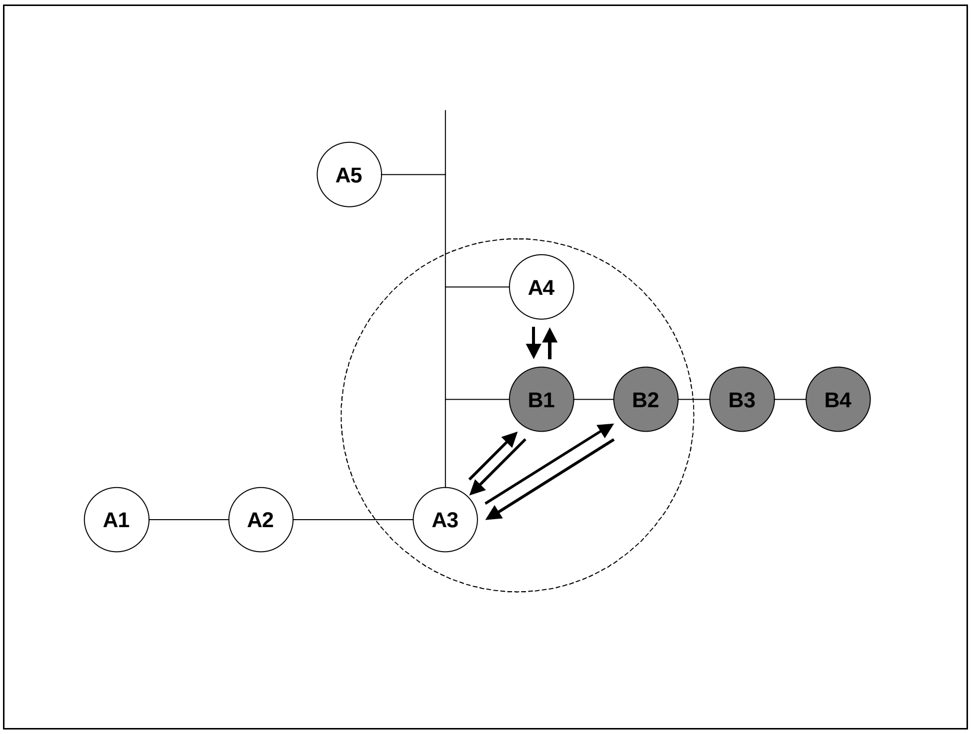

Figure 82 Example of basic interference

In this basic example, the coexistence issues are:

- A3 may cause interference to B1 and B2
- A4 may cause interference to B1
- B1 may cause interference to A3 and A4
- B2 may cause interference to A3

If all nodes use the same frequencies at the same time :

- the receptions of A3 and A4 are disturbed by the transmissions from B1
- the reception of A3 is disturbed by the transmissions from B2
- the reception of B1 is disturbed by the transmissions from A3 and A4
- the reception of B2 is disturbed by the transmissions from A3

B.1.2 Purpose

The purpose of the specification is to provide mechanism to avoid frame collisions between different systems that share parts of the same electrical installation.

This specification fulfills the following requirements:

- Work with the typical topologies in the world.
- Provide a fair and balanced sharing of resources for up to 3 In-home systems that might otherwise interfere locally.
- Provide a fair and balanced sharing of resources between the Access system and the In-home systems when both systems coexist - 50% of the channel resources available for the Access system and 50% to be shared between the In-home systems.
- Avoid interference between neighboring systems thus minimizing loss of throughput.
- Allow the Access system to reuse the channel resources when In-home systems are inactive for more than 5 minutes.
- Allow the In-home system to reuse the channel resources when Access systems are inactive for more than 10 minutes.

- Do not decrease the original throughput of each system by more than 5 % when alone.

- Allow one Access system operating on a dedicated frequency sub-band and coexisting on the other sub-band with 3 In-home systems to operate with a typical cell channel access latency less than 38 ms.

- Allow one Access system, when it does not operate on a dedicated frequency sub-band, coexisting with 3 In-home systems to operate with a typical cell channel access latency less than 13 ms.

- Allow one Access system and 3 In-home systems coexisting to operate with a typical cell channel access latency less than 65 ms for the In-home systems.

- Allow one Access system and 3 In-home systems coexisting to operate with maximum cell channel access jitter of 12.5ms for the Access system and with the maximum of 80 ms for 3 In-home systems.

- Do not increase the size of the components (analogue and digital) by more than 5% with any of the existing technology of broadband transmission.

- Work with Access systems deployed with time repeaters or frequency repeaters.

- Optimize the use of resources in time and frequency domains in order to achieve the optimum performance for each of the coexisting systems.

- Resource sharing rules shall be locally applied.

- Coexistence mechanism should induce no extra EMC radiation levels.

- Booting time should not be increased by more than 5 seconds due to coexistence mechanism.

B.1.3 Current Status

This document contains the first version of the UPA's proposal for Coexistence Specification. This version is being released to solicit feedback and to open dialog on the topic. All values proposed in this document are to be considered preliminary and need to be validated with further testing.

In addition, there are some additional issues that need further investigation. One of the open issues is regarding the probabilities of collisions occurring and a mechanism to recover from such collisions. The current expectation is that the mechanisms provided in this specification provide for greatly reducing the probabilities of collisions occurring. However, some probability of collisions is introduced by the complex and constantly changing nature of the power line channel. Conditions such as these will be further investigated and provided for in the future release of the specification.

The UPA expects to release a second version within ninety days of the release of this version.

B.2 Definitions

node: a node is an access point to the medium

device: a device is made of a single node except frequency repeaters.

system: a system is made of devices that communicate between each other to provide services to users

cell: a group of nodes of a same system that communicate between each others using the same frequency band

frequency repeater: a device made of two nodes belonging to two different system cells

access system: a system that delivers external services, such as Internet, VoIP, IPTV, utility related services, etc.

In-home system: a system that sources and delivers services within the home, like AV & LAN extensions; In-home systems may be connected to an Access system to extend external services. An In-home system is made is of single cell.

cell channel access latency: it is the time for a cell to get authorization to transmit

cell channel access jitter: it is the standard deviation of the cell channel access latency

coexistence signals: SyncSig and PresSig signals

B.3 Abbreviations and acronyms

AS	Access system
FB	frequency band
IHS	In-home system

B.4 Operating frequency bands

A system cell is said to operate in the frequency band FB1 if it only makes use of frequencies below 12 MHz.

A system cell is said to operate in the frequency band FB2 if it only makes use of frequencies above 13 MHz.

A system cell is said to operate in the frequency band FB3 if it both makes use of frequencies below 12 MHz and above 13 MHz.

Important note: The frequency bands are given for information. They will be definitively defined after validation tests.

B.5 Detection of interference

The detection of interference is based on the detection and transmission by the nodes of two un-modulated coexistence signals:

- SyncSig: Coexistence signal used to synchronize the nodes
- PresSig: Coexistence signal used to notify the presence of nodes, manage cell ID selection, detect cell ID conflicts and manage cell resynchronizations.

B.5.1 Notification of presence

Each node shall send periodically a SyncSig signal.

Each node from the same cell shall synchronize their SyncSig signals as defined in section B.5.2.

To notify their presence, each node sends some PresSig signals after a specific delay from the SyncSig signal as described here-below and in sections B.5.2 and B.5.3. After the SyncSig signal in time, a Presence Window is defined with eleven mini-slots of time.

Figure 83 Notification of presence

- The nodes from an Access cell operating in the frequency band FB1 shall send the PresSig signal every SyncSig period during the mini-slot A.
- The nodes from an Access cell operating in the frequency band FB2 shall send the PresSig signal every SyncSig period during the mini-slot B.
- The nodes from an Access cell operating in the frequency band FB3 shall send the PresSig signal every SyncSig period during the mini-slot C.
- All nodes belonging to the same In-home cell shall send the PresSig signal the same mini-slot selected from mini-slot 1, 2 or 3 as described in section B.5.3.

The S mini-slot is used to control the re-synchronization of nodes (see section B.5.2)

The X mini-slot is used for the cell ID management (see section B.5.3)

The three last mini-slots are reserved for future use.

B.5.2 Node synchronization and detection of synchronization conflicts

From a synchronization point of view, a node has three states: unsynchronized, re-synchronizing and synchronized.

For the purpose of this chapter, the following definitions are given:

- Valid sequence of periodic SyncSig signals: a set of three consecutive 153.6ms SyncSig signals. SyncSig signals which are received within less than 153.6ms correspond to another node synchronization and might be part of another valid sequence of periodic SyncSig signals.

- Access sequence* : a set of three consecutive 153.6ms SyncSig for which the associated presence window notifies an Access presence (it can also include an in-Home notification).

- Unidentified sequence*: a set of three consecutive 153.6ms SyncSig for which no Access presence is detected in the associated presence window.

- Sequence under synchronization: a set of three consecutive 153.6ms SyncSig for which an S-signal is notified in the associated presence window.

*__Note__: An Access node is always required to notify its presence by transmitting its signal into the presence window whereas an in-Home Node does not follow the same constraint (see Section B.5.3) Therefore, detection of an Access sequence is possible within a valid sequence of periodic SyncSig signals whereas detection of the exact ID of an in-Home sequence is not guaranteed.

Unsynchronized state:

Initial state: An unsynchronized node shall sense the channel looking for SyncSig signals.

1. If an unsynchronized node does not sense any valid sequence of periodic SyncSig signals for more than 1 second, it shall enter the synchronized state.

2. If an unsynchronized node receives SyncSig signals among which only one valid sequence of 153.6ms periodic SyncSig signals can be detected, the node shall lock onto this sequence and shall listen for the S-mini slot of the presence window attached to this SyncSig.

    a. If an S-signal is detected (sequence under synchronization) the node shall reinitiate the unsynchronized state.
    b. If no S signal is received in the presence window, the node should enter the synchronized state.

3. If an unsynchronized node hears SyncSig signals among which several distinct valid sequences of 153.6ms periodic SyncSig signals can be detected, the node should evaluate the PresSig signal of each sequence. Different situations might involve different state transitions:
    a. At least one of the sequences is not under resynchronization:
        i. If an Access sequence is detected:
            1. If there is at least one Unidentified sequence not under resynchronization, the node should lock onto one of this (these) sequence(s) and enter the re-synchronizing state.
            2. If there is only one Access sequence not under resynchronization whereas all the other sequences are under resynchronization, the node should lock onto this Access sequence and enter the synchronized state.

3. If there are several Access sequences not under resynchronization and all the Unidentified sequences are re-synchronizing, the node should lock onto one of the Access sequence and enter the re-synchronizing state.

ii. If no Access sequence is detected:

1. If there is only one single Unidentified sequence not under resynchronization, the node should lock onto this sequence and enter the synchronized state.
2. If there are at least two Unidentified sequences not under resynchronization, the node should lock onto one of the Unidentified sequences and enter the re-synchronizing state.

b. All the sequences are in the re-synchronizing state, the node shall enter the unsynchronized state.

Re-synchronizing state:

A re-synchronizing node is a node which is synchronized to a cell to be resynchronized.

Initial state: A node which enters the re-synchronizing state shall maintain its current synchronization for a 5 second period or until the valid sequence of periodic SyncSig signals is finished (whichever is shorter),, echoing the S notification in the presence window. A re-synchronizing node is not required to transmit the SyncSig signal. It is also not required to transmit a cell ID.

Once the S-echo period is over, the re-synchronizing node should then enter back the unsynchronized state.

Synchronized state:

A synchronized node shall periodically send the SyncSig signal and listen for an S signal notification in the presence window.

As soon as the node detects an S signal, it shall enter the re-synchronizing state.

An Access node is required to always transmit its PresSig signal into the presence window.

An in-Home node which enters or runs the synchronized state shall permanently run the Cell ID Management process. In particular, the Cell ID Management process defines the transmission requirements of the PresSig signals.

B.5.3 Cell ID management

The Cell ID Management (CIM) process only applies to in-Home synchronized nodes. The CIM process carries the following functions:

1. **Cell ID selection**: All the nodes within the same system have to choose the same cell ID. Moreover, this ID has to be the lowest unused ID.

2. **Cell ID saturation notification and management**: When a new node, not part of any system detected on the channel senses all Cell IDs being used, it will issue a Cell ID saturation notification marking the X signal in the presence window. This mechanism provisions for scenarios where more than three in-Home systems with direct visibility try to coexist. Under these circumstances, every node should switch to a default Factory Cell ID (1, 2 or 3). Factory Cell IDs are provided by UPA to different technology providers implementing this specification so that all modems of the same technology have the same factory ID. When the saturation scenario is detected, the coexistence rules apply to technologies (Factory Cell IDs) instead of systems (Cell IDs). Therefore, every technology should provide its own mechanisms to coexist with different networks of its own technology in order to share the resources.

3. **Cell ID collision detection**: two synchronized nodes of different systems sensing each other presence signals cannot have the same ID. A node shall never be able to sense its own Cell ID on the channel. When a node senses its own Cell ID, it means a Cell ID collision. This requires that all nodes part of the same system be either in transmission or reception mode at the same time during the presence mini-slots associated to their Cell ID.

4. **Cell ID collision resolution**: After a Cell ID collision is detected, only one of the colliding systems shall change its Cell ID. The Cell ID Change Management algorithm solves this issue.

5. **Cell ID change notification and management**. If a node detects that a system with a lower Cell ID has been turned off (no Cell ID transmissions detected for more than 5mn), or if it is notified of a Cell ID saturation via the X mark or the Cell ID collision resolution algorithm it should communicate the Cell ID change to the other nodes in the system. Use of lower possible Cell ID is important to maximize the spatial reuse.

Note: These functions require vendor specific implementations.

B.5.3.1 Description of the CIM phases:

For the purpose of this chapter, the following definitions are used:

Coexistence frame: A 153.6ms frame starting with the SyncSig signal.

Cell ID sequence: A set of M consecutive coexistence frames (The value of M is defined in Table 39).

Alive Cell ID: A Cell ID which has been sensed within the last 5 mn. The list of Alive Cell IDs is initially null when booting and gets updated after each Cell ID sequence.

Dead Cell ID: A Cell ID which has not been sensed for more than 5mn.

Systems with direct visibility: A set of systems in which some nodes can sense each other Cell ID presence signals.

Data slot sensing (booting task):

This phase applies to booting nodes which just completed synchronization phase (section B.5.2). It is evaluated before the Cell ID sensing phase is started (or completed in parallel). It guarantees that two nodes of the same system do not select different Cell IDs.

In this phase, the node senses the data slots of the received coexistence frames.

1. If the data slots are unintelligible, the node is not part of any of the systems on the channel. Go to the Cell ID sensing phase.

2. If some data slots are intelligible, the node tries to connect to those systems whose data are intelligible to see if it belongs to any of them.
    a. If the node does not form part of any of the systems present, it goes to the Cell ID sensing phase.
    b. If the node forms part of one of the systems present, the node selects the Cell ID of the system it belongs to (see note 1) and enters the Cell ID sensing phase with a Cell ID already selected.

Note1: This mechanism requires a vendor-dependent implementation for the node to get the notification of appropriate Cell ID used by its system pairs.

Cell ID sensing (background task):

This phase is a background task, permanently executed by synchronized nodes.

It is based on the evaluation of all the signals received in the mini-slots 1,2 3 and X, sensed during a Cell ID sequence. It leads to the selection/confirmation of the Cell ID.

The following situations must be evaluated

1. If a saturation notification is detected (at least 3 X signals detected in the Cell ID sequence), the node should switch to its Factory Cell ID and echo the X signal for the next 3 Cell ID sequences.

2. If no saturation notification is detected:
    a. If the node has no Cell ID it should always select the lowest available Cell ID:
        i. If no Cell ID is alive, the node should select Cell ID 1.
        ii. If only Cell ID 1 is alive, the node should select Cell ID 2.
        iii. If Cell ID 2 and Cell ID 1 are alive and no Cell ID 3 is alive, the node should select Cell ID 3.
        iv. If Cell ID 1, 2 and 3 are alive, the node should issue an X signal for the next three Cell ID sequences.
    b. If the node already has a Cell ID:
        i. If it detects a Cell ID which is equal to its own Cell ID (see Cell ID Collision Detection), go to the Cell ID Change Management Phase.

ii. If one of the alive Cell IDs becomes dead, and this dead Cell ID value is greater than the existing alive Cell IDs, go to the Cell ID Change Management phase.

Cell ID Collision Detection

A Cell ID collision occurs when two or more nodes belonging to different cells with direct visibility select the same Cell ID. Depending on the attenuation level, Cell ID collision may result in the loss of transmitted data, and therefore collisions should be avoided. In order to detect Cell ID collisions, all nodes within one cell should either be in transmission or in reception mode during the Presence Window. If a node that is in reception mode during the Presence Window senses its Cell ID, a cell ID collision has then been detected.

The probability of having a Cell ID collision depends on the state of the channel, and it is difficult to model. Therefore, this specification deals with the maximizing the probability of detecting such collision. In order to maximize the probability of the detection of a Cell ID collision, the sequence of transmission/detection mode within the Presence Window should be based on a random process and it should not be periodic. The method by which the sequence of transmission/detection mode is notified to the nodes in the network, and the random algorithm are left to vendor implementation, although there are three parameters that have to be maintained in all implementations:

- The period of changing the transmission/reception sequence (M).
- The number of times within a Cell ID sequence that the Cell ID should be transmitted by every cell (N).
- The number of times a cell has to detect its Cell ID within a Cell ID sequence to decide that there has been a collision (L).

With these constraints, the probability of detecting a Cell ID collision within one Cell ID sequence of length M given than two cells have selected the same Cell ID is:

$$P(D) = P(X >= L) = \sum_{i=L}^{\min\{M-N,N\}} P(X = i) = \sum_{i=L}^{\min\{M-N,N\}} \frac{\binom{M-N}{i}\binom{N}{N-i}}{\binom{M}{N}}$$

Equation 46 Probability of Cell ID Collision Detection

Equation 46 shows the probability of detecting a cell ID collision as it was defined in the previous paragraph during one Cell ID sequence of length M. X is a random variable that indicates the number of times a Cell ID is detected in a Cell ID sequence of length M. Therefore, the probability of detecting a collision is only possible when the random variable X is greater or equal to L.

	Value

M	15
N	8
L	3
P(D)	0.899

Table 39 Cell ID Collision Detection

With the values shown in Table 39, the number of coexistence frames sequences required to be transmitted to have a probability above 0.99 to detect the collision is two.

Cell ID Collision Resolution

Once a Cell ID collision has been detected, all nodes in the cell that detected the collision are notified and a new Cell ID should be selected using Cell ID Change Management.

Cell ID Change Management

A Cell ID may change due to three reasons:

1. **Lower Cell ID becomes available**. A Cell ID becomes available after 5 minutes since the last time it was detected plus some offset time that depends on the current Cell ID. There are two possible scenarios:
    a. If the current Cell ID is 3, the offset time is 0.
    b. If the current Cell ID is 2, the offset time is 1 extra minute.

    Once a lower Cell ID becomes available, the cell that detected the Cell ID available, transmits both Cells IDs in the Presence Window (old and new one), although it keeps transmitting only during the old Cell ID slots, and maintains the same transmission/detection sequence for both Cell IDs.

    If during 3 Cell ID sequences, a Cell ID collision with the new Cell ID has not been detected, the cell selects the new Cell ID as the current Cell ID. On the other hand, if a collision has been detected with the new Cell ID, the cell maintains the old Cell ID as the current Cell ID.

2. **Cell ID collision resolution**. A first cell that detects a collision should reset all information regarding used Cell IDs and proceed to the Data Slot Sensing phase.

3. **X signal detection**. As soon as an X signal is detected, every node in the cell should change its Cell ID to the factory default ID.

B.5.4 Detection of disturbed and disturbing nodes

When an Access node detects in the Presence Window a PresSig signal in mini-slot 1, 2 or 3, it means that it has received a PresSig signal from an In-home cell and that it could be disturbed by a node from a neighboring In-home cell. Then the node records the names of the mini-slots in which it detected a PresSig signal and changes its state to Disturbed node.

When an In-home node detects in the Presence Window a PresSig signal out of its own mini-slot, it means that it receives a PresSig signal from another cell (Access or In-home) and that it could be disturbed by a node from a neighboring cell. Then the node records the names of the mini-slots in which it detected a PresSig signal and changes its state to Disturbed node.

By reciprocity, we assume that a Disturbed node may disturb some nodes in neighboring cells. Automatically, the Disturbed nodes also get the state of Disturbing node.

When a node stops detecting a PresSig signal from an In-home cell in mini-slots 1,2 or 3 for more than 5 minutes, or from an Access cell in mini-slots A,B or C for more than 10 minutes,, the node clears the name of the mini-slot from its records and updates both states of Disturbed and Disturbing nodes accordingly.

B.5.5 Notification of disturbed and disturbing nodes to the other nodes of the cell

To avoid collisions, a node that wants to transmit data to another node shall know if the destination node is a Disturbed node or not and from which neighboring cell it is disturbed.

When a node gets the state of Disturbed node or when its record list of detected mini-slots changes or when a Disturbed node loses its status of Disturbed node, the node shall inform within less than one second the other nodes that may want to communicate to that node of its new record list of detected mini-slots and its new status. The nodes of the cell should keep an updated list of the Disturbed nodes in the cell with their detailed record lists of detected mini-slots.

The signals and messages used by a node to inform the other nodes of its cell with its detailed status of Disturbed node, are not described in this specification. It is only required to the systems to do it, by their own means of communication.

B.5.6 Signals Used

This section specifies the physical layer characteristics of the Coexistence signals used for the detection of possible interference. This PHY is proposed because of its inherent adaptability in the presence of frequency selective channels, its resilience to jamming signals, its robustness to impulsive noise, its capacity for achieving high spectral efficiencies and its low implementation complexity.

B.5.6.1 Equations

The description of both signals is based on a 40 MHz clock and they are composed of OFDM symbols that may be generated with a complex 256-point IFFT repeating the sequence four times, as shown in Equation 1.

$$s_{D,i}(n) = k_D w_D(n) \left\{ \sum_{k=1}^{N_D/2} c(k) \exp\left(\frac{j2\pi k}{256} n \right) + \sum_{k=255-N_D/2}^{255} c(k) \exp\left(\frac{j2\pi k}{256} n \right) \right\}$$

Equation 47 OFDM symbols

Where

kD is a normalization factor

- wD(n) is a windowing function
- ND is the number of active subcarriers.
- c(k) = exp(jϕk) where ϕk is a random phase picked from a uniform distributed random variable between [0,2π[

Four of these symbols are concatenated to form the signals.

$$s_{PresSig}(n) = \sum_{i=0}^{3} s_{D,i}(n - 256i)$$

$$s_{SyncSig}(n) = \sum_{i=0}^{3} (-1)^i s_{D,i}(n - 256i)$$

Equation 48 Coexistence signals

B.5.6.2 Frequency ranges

See Table 2 in section B.6.1.

B.5.6.3 Power spectrum densities

A node transmits the coexistence signals at the output power level it uses to transmit data – 5 dB.

A node sets its threshold for the detection of the coexistence signals at its sensitivity level + 20 dB.

B.5.7 Timings

The SyncSig signal shall be transmitted once every 153.6 msecs.

The length of the SyncSig signal is of 300 µs.

The Presence Window is made of eleven mini-slots of 300 µs each.

B.6 Collision avoidance mechanism in frequency and in time

The following sections describe the mechanism to avoid collisions in case of:

- interference between an Access cell and an In-home cell
- interference between two In-home cells

Note : when a cell does not detect any interference, for example a single In-home system operating on a electricity installation, the collision avoidance mechanism is not activated neither in frequency nor in time. Therefore, the nodes of the cell shall only transmit the SyncSig and PresSig signals as described in section B.5 and are allowed to transmit data anytime in between.

B.6.1 Collision avoidance between an Access cell and an In-home cell

The following table specifies the rules to avoid collisions when an interference is detected between an Access cell and an In-home cell.

Interference between an Access cell and an In-home cell is detected when the cells contain at least a Disturbed Node or a Disturbing Node.

In the table 2 here-above,

- the title row represents the possible operating frequency bands of an In-home cell when it does not interfere with any other cells.
- the title column represents the possible operating frequency bands of an Access cell when it does not interfere with any other cells.
- the boxes represent the actions to take when interference between the two cells is detected.

	IHS cell operating in FB1. The IHS nodes transmit and listen to the coexistence signals between 2 and 12 MHz.	IHS cell operating in FB2. The IHS nodes transmit and listen to the coexistence signals between 13 and 30 MHz.	IHS cell operating in FB3. The IHS nodes transmit and listen to the coexistence signals between 2 and 30 MHz.
AS cell operating in FB1. The AS nodes transmit and listen to the coexistence signals between 2 and 12 MHz.	**Collision avoidance using frequency division.** - The IHS Disturbing Nodes switch to FB2. - The IHS nodes communicating to Disturbed Nodes switch to FB2. The IHS nodes transmit the coexistence signals between 13 and 30 MHz. The IHS nodes listen to the coexistence signals both between 2 and 12 MHz and 13 and 30 MHz.	None. No interference should be detected.	**Collision avoidance using frequency division.** - The IHS Disturbing Nodes switch to FB2. - The IHS nodes communicating to Disturbed Nodes switch to FB2. The IHS nodes transmit the coexistence signals between 13 and 30 MHz. The IHS nodes listen to the coexistence signals between 2 and 30 MHz.
AS cell operating in FB2. The AS nodes transmit and listen to the coexistence signals between 13 and 30 MHz.	None. No interference should be detected.	Collision avoidance using time division. The transmissions to the Disturbed Nodes from the Disturbing Nodes are scheduled in time.	**Collision avoidance using time division.** The transmissions - to the Disturbed Nodes - from the Disturbing Nodes are scheduled in time.
AS cell operating in FB3.	Collision avoidance using time division.	**Collision avoidance using time division.**	**Collision avoidance using time division.**

The AS nodes transmit and listen to the coexistence signals between 2 and 30 MHz.	The transmissions - to the Disturbed Nodes - from the Disturbing Nodes are scheduled in time.	The transmissions - to the Disturbed Nodes - from the Disturbing Nodes are scheduled in time.	The transmissions - to the Disturbed Nodes - from the Disturbing Nodes are scheduled in time.

Table 40 Collision avoidance protocol between an Access cell and an In-home cell

B.6.2 Collision avoidance between two In-home cells

When an interference is detected between two In-home cells, collisions are avoided using time division. The transmissions to the Disturbed Nodes and from the Disturbing Nodes are scheduled in time.

B.6.3 Collision avoidance between more than two cells

If a cell interferes with more than one other cell, the collision avoidance rules defined in sections B.6.1 and B.6.2 are accumulated.

B.6.4 Time division protocol

Reminder. This section does not apply to the coexistence with Access cells operating in the Frequency Range FB1 that is addressed by frequency division.

After the Presence Window, a Data Window is defined with twelve slots of time numbered from S0 to S11.

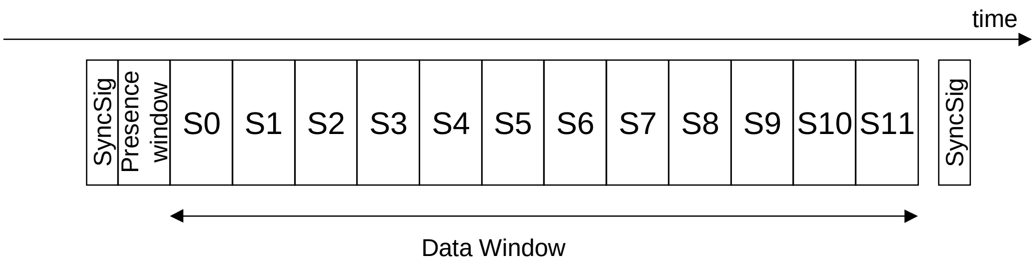

Figure 84 Time Slots slicing

To avoid collisions, the slots are pre-assigned to the systems cells as described in the following table.

Cell	Pre-assigned slots
Access #B and #C	- S0, S4 and S8, if an Access cell #A is detected - Even slots (S0, S2,…, S10), else.
In-home #1	S1 and S7
In-home #2	S3 and S9
In-home #3	S5 and S11

Table 41 Pre-assigned time slots to avoid collisions

B.6.4.1 Protocol for Access cells

For each given cell, the rules to avoid collisions are:

- a Disturbing node shall transmit during the time slots pre-assigned to its own cell.
- a node that wants to transmit towards a Disturbed node should transmit during the time slots pre-assigned to its own cell.
- in the other transmission cases, a node is free to transmit any time during the Data Window.

B.6.4.2 Protocol for In-home cells

For each In-home cell, the rule to avoid collisions are:

- a Disturbing node shall not transmit during the time slots pre-assigned to the cells its own cell is interfering with. (As reminder, every node of a cell knows of other cells with which its own cell interferes. See section B.5.5).
- when a Disturbing node chooses to transmit during a time slot not pre-assigned to any cell its own cell is interfering with, the node shall contend with the nodes from the other cells to gain the right to use it. These free time slots are named dynamic time slots. The contention mechanism is defined in the section B.6.4.3.
- a node that wants to transmit towards a Disturbed node should not transmit during the time slots pre-assigned to the cells its own cell is interfering with. (As reminder, every node of a cell knows of other cells with which its own cell interferes. See section B.5.5).
- when a node that wants to transmit towards a Disturbed node chooses to transmit during a time slot not pre-assigned to any cell its own cell is interfering with, the node shall contend with the nodes from the other cells to gain the right to use it. These free time

slots are named dynamic time slots. The contention mechanism is defined in the section B.6.4.3.

- in the other transmission cases, a node is free to transmit any time during the Data Window.

B.6.4.3 Contention mechanism for using the dynamic time slots

The contending In-home cells dynamically adjust their reservation priority depending on the last time they reserved a dynamic slot. Thus, dynamic slots are distributed fairly among contending In-home cells.
Reservation of dynamic slots is done transmitting a PresSig signal. If a cell contending for the reservation of a dynamic slot detects a PresSig in any of the mini-slots that precede the dynamic slot, it will defer the contention until the next dynamic slot, varying its reservation priority. This method ensures that the cell with lowest reservation ID will win the contention.

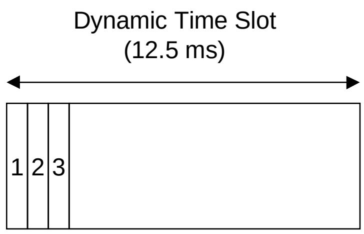

Figure 85 Contention window within dynamic time slots

In Figure 4, the structure of the dynamic slot is depicted. Positions 1, 2 and 3 are used to transmit the PresSig to reserve the slot. The rest of the dynamic slots are used for data transmission.
An In-home cell that wants to use the dynamic slot shall transmit the reservation signal in the adequate mini-slot of the dynamic slot to reserve the dynamic slot. The reservation signal position (1, 2 or 3) is selected using the following algorithm:

1. Each cell initially uses its own ID value for the reservation window. As reminder, the cells that interfere have chosen different IDs (see section B.5.3)
2. When a cell reserves the dynamic slot, it resets the reservation value to the number of In-home cells that it interferes with plus 1 (that is, counting itself).
3. When a cell that contends for a dynamic slot detects a reservation signal reserving the slot, it defers contention until next dynamic slot, and subtracts one from the reservation value.

Note: The above algorithm optimizes the fair reuse of dynamic slots between different coexisting and contending systems (see Scenarios described in Section B.7). However, it must be noted that this reservation mechanism is not collision proof.

B.6.4.4 Timings

The Data Window is made of twelve time slots S0 to S11 of 12.5 ms each.

The three mini-slots 1,2 and 3 used for contention within time slots are of 300 µs each.

B.7 Illustrations

B.7.1 General coexistence scenarios

Note: in this section, the notation A, B, C and D has no connection with the name of the mini-slots A, B, and C used in section B.5.

There are three basic scenarios where coexistence among cells is required. There can be other topologies resulting from a combination of these three scenarios.

The first scenario is shown in Figure 86. This topology represents a cell that has visibility with several other cells. However, the other cells do not see each other. In this case, Cell A has to coordinate its transmission with the other three cells.

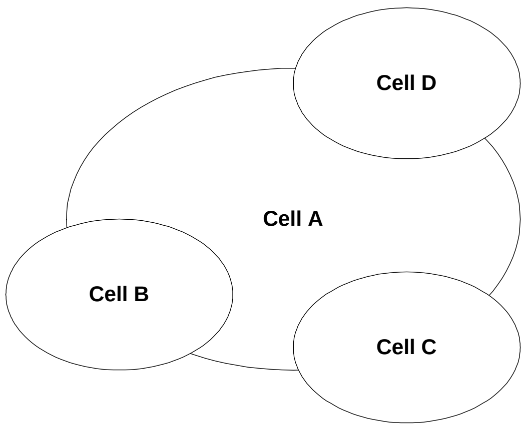

Figure 86 Topology I

Figure 87 depicts the expected transmission schedule for this scenario. Since cell A sees the other three cells, it has to coordinate its transmission slot with the other three cells. As it can be seen, the transmission of cell A and the other cells are alternated. Since cells B, C and D do not see each other, they can transmit simultaneously.

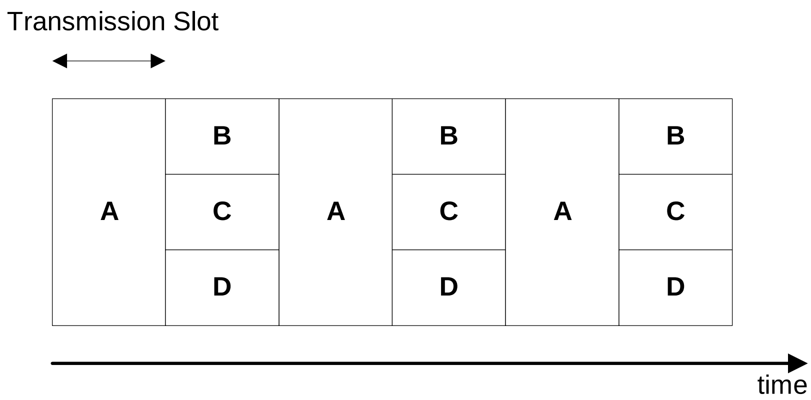

Figure 87 Transmission Schedule Topology I

Figure 88 represents a linear topology. In this scenario, there is a number of cells, but the visibility among them is limited to neighboring cells. The transmission schedule in this scenario should look something like the schedule shown in Figure 89, where independent cell transmissions are coordinated in order not to interfere with each other.

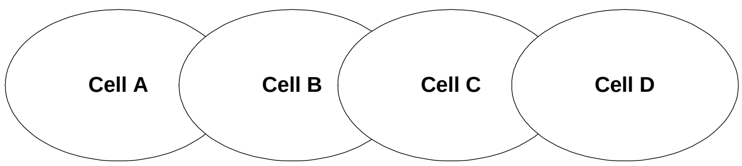

Figure 88 Topology II

Figure 88 shows a scenario where it is important having fixed length transmission slots so that synchronization can be achieved. If transmission slot length is not fixed, cell B will be blocked, because either cells A or C will occupy the channel. If slots are fixed in length, transmissions can be synchronized to yield maximum spatial reuse.

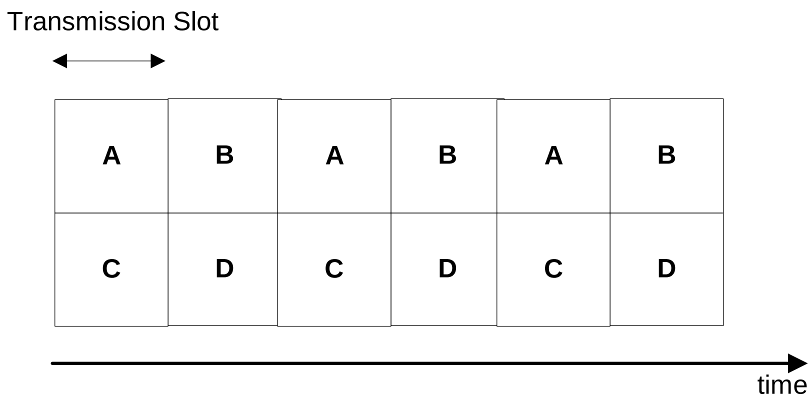

Figure 89 Transmission Schedule Topology II

Finally, in the last scenario, all cells have visibility of each other. An example of this scenario is shown in Figure 90 and the transmission schedule is shown in Figure 91.

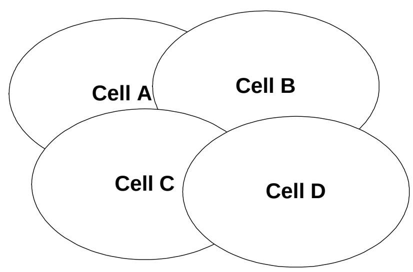

Figure 90 Topology III

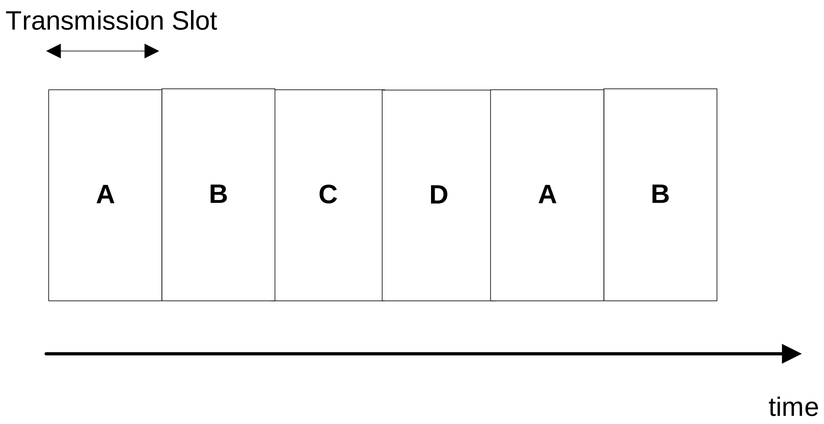

Figure 91 Transmission Schedule Topology III

This example shows the minimum latency that can be guaranteed with such a coexistence method.

B.7.2 Illustrations of collision avoidance mechanism

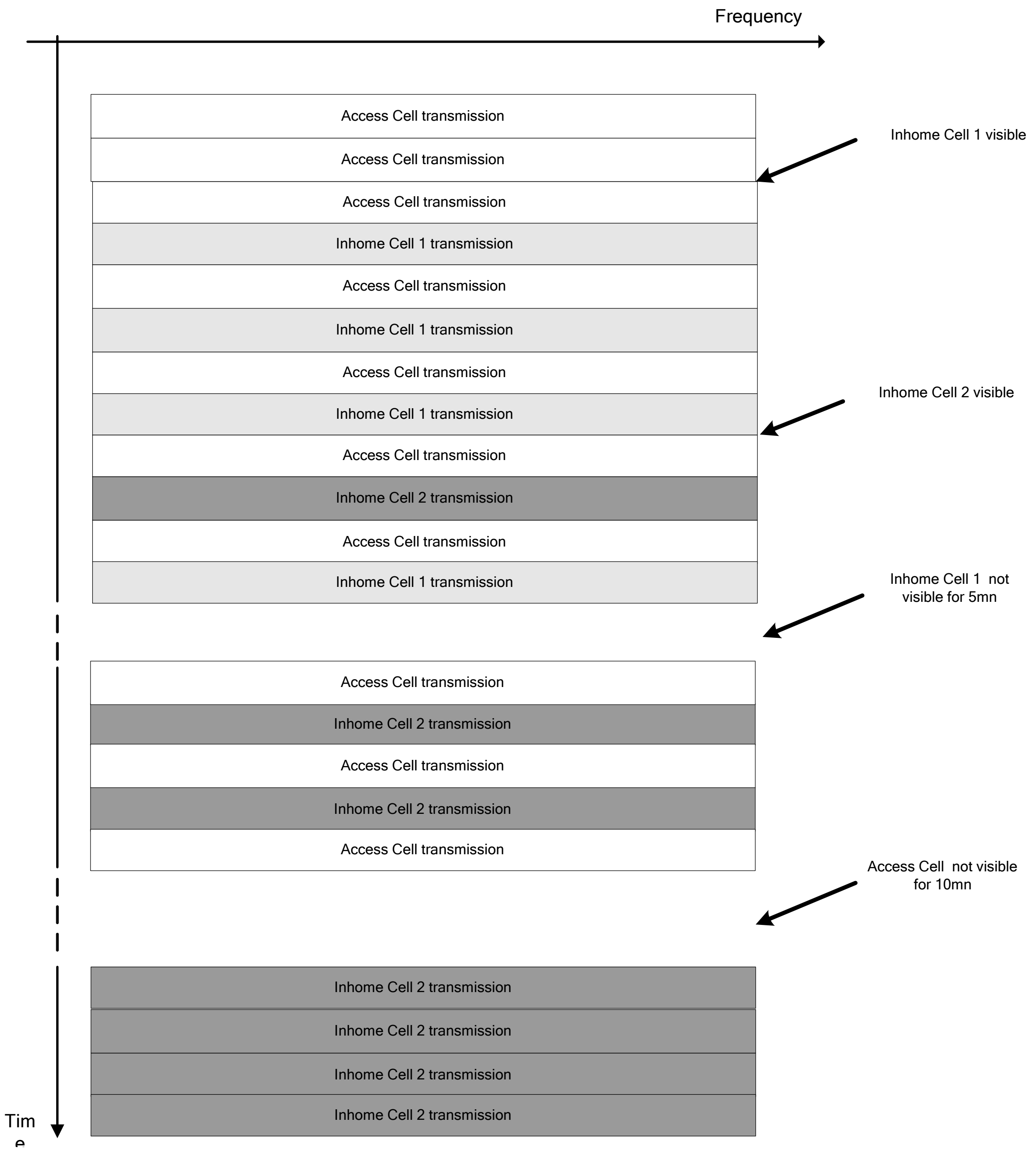

Figure 92 Example of coexistence between an Access cell and In-home cells both operating in FB3

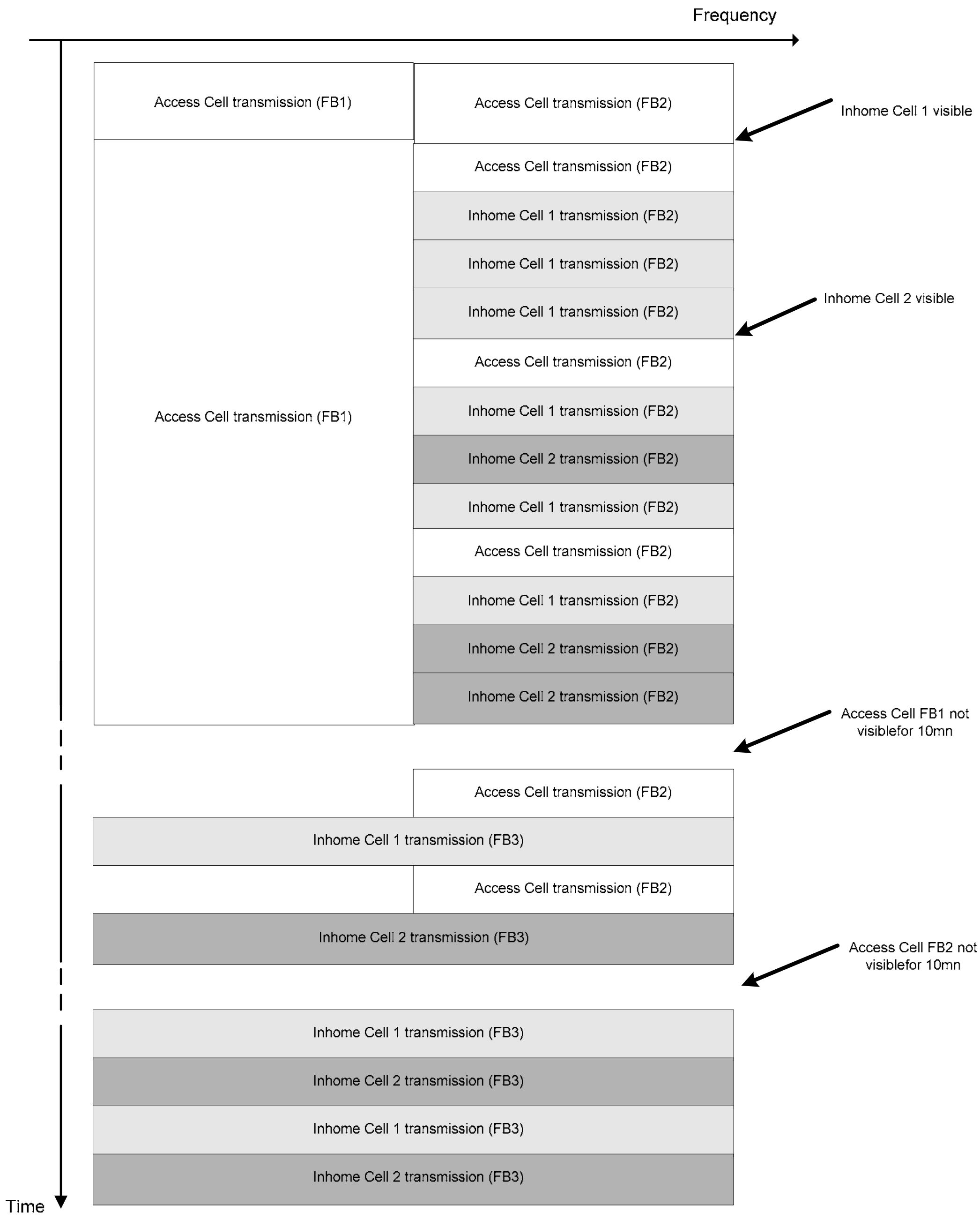

Figure 93 Example of coexistence between two Access cells operating in FB1 and FB2 and two In-home cells operating in FB3

B.7.3 Detailed scenarios

In this section, several coexistence scenarios are analyzed to gain some insight on how the proposed coexistence mechanism works. Notice that in all figures in this scenario, it is assumed that cells always transmit the Cell ID. In reality, Cell ID transmission follows the rules detailed in section B.5.3.

B.7.3.1 Scenario A. Two In-home cells operating in FB2 or FB3.

In this scenario A, only two In-home cells coexist on the same electrical installation. It is assumed that the cells took the IDs #1 and #2.

Nodes in cell 1 detect cell 2 because of the signal in the presence window 2. The same is true for cell 2. Cell 1 has slots 1 and 7 pre-assigned, while cell 2 has slots 3 and 9 pre-assigned. The remaining time slots will be dynamically assigned.

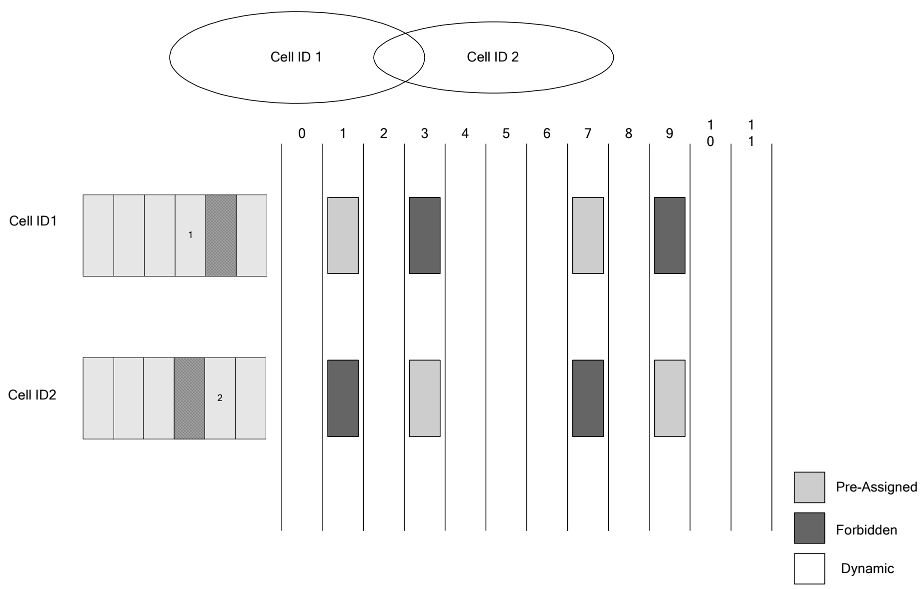

Figure 94 Example of use of pre-assigned slots

The figure 13 illustrates the use of the pre-assigned time slots in the cells. Nodes in cells #1 and #2 are contenting for the use of the dynamic time slots, i.e the slots pre-assigned to In-home cells #3 and Access cells.

Dynamic slots are reserved using the reservation protocol described in section B.6.4.3. The way to reserve a dynamic slot is by transmitting a PresSig. If a node detects a PresSig transmitted by other cell, it has to defer contention until next dynamic slot. Contending cells select a different reservation number, as specified in Section B.5.3. This mechanism allows fairness among all contending cells that implement the reservation mechanism.

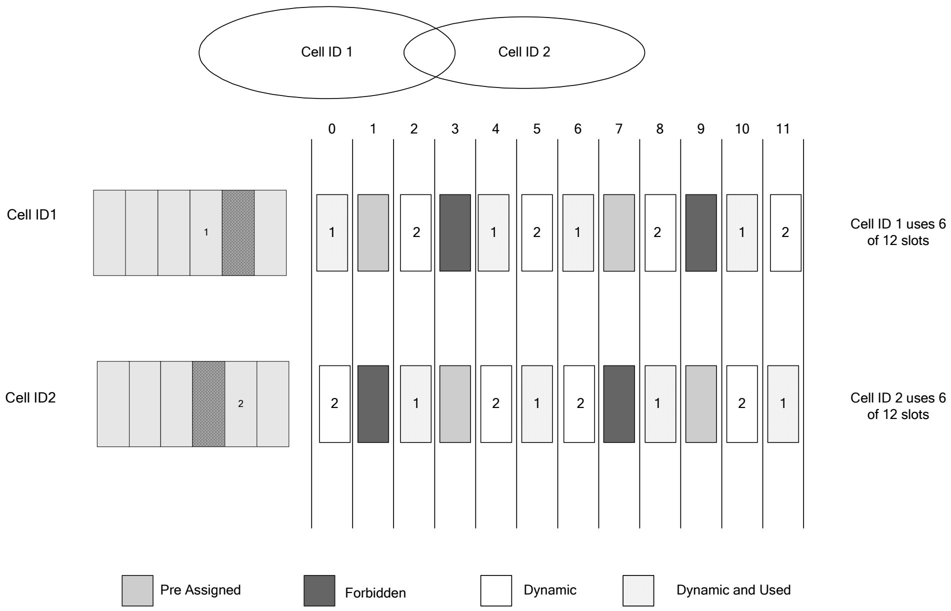

Figure 95 example of allocation of dynamic time slots

Figure 14 shows how dynamic slots are assigned in the two-cell scenario. Dynamic slots are slots 0,2,4,5,6,8,10 and 11. So, at the beginning of the frame, cell 1 sets its reservation ID to 1, and cell 2 sets its reservation ID to 2. The first dynamic slot (slot 0) is won by cell 1, because its reservation ID is lower. Since cell 1 won the contention, it sets its reservation ID to 2. On the other hand, cell 2 lost the contention and sets its reservation ID to 1. The next dynamic slot is slot 2. The contention is won by cell 2 because its reservation ID is lower. Following the way the slots are assigned, cell 1 uses slots 0,1,4,6,7 and 10 (six slots), whereas cell 2 uses slots 2,3,5,8,9 and 11 (six slots).

B.7.3.2 Scenario B. Four In-home cells operating in FB2 or FB3.

Scenario B shows how the coexistence mechanism works for more complex scenarios, where not all cells detect the presence of the rest. In this scenario, the dynamic slots will be used simultaneously by different cells with no visibility, providing spatial reuse.

A cell always selects the lowest ID available, so cells that do not see each other select the same ID. In the next example depicted in Figure 16, two cells have selected the 1 ID (one will be marked as 1a and the other as 1b):

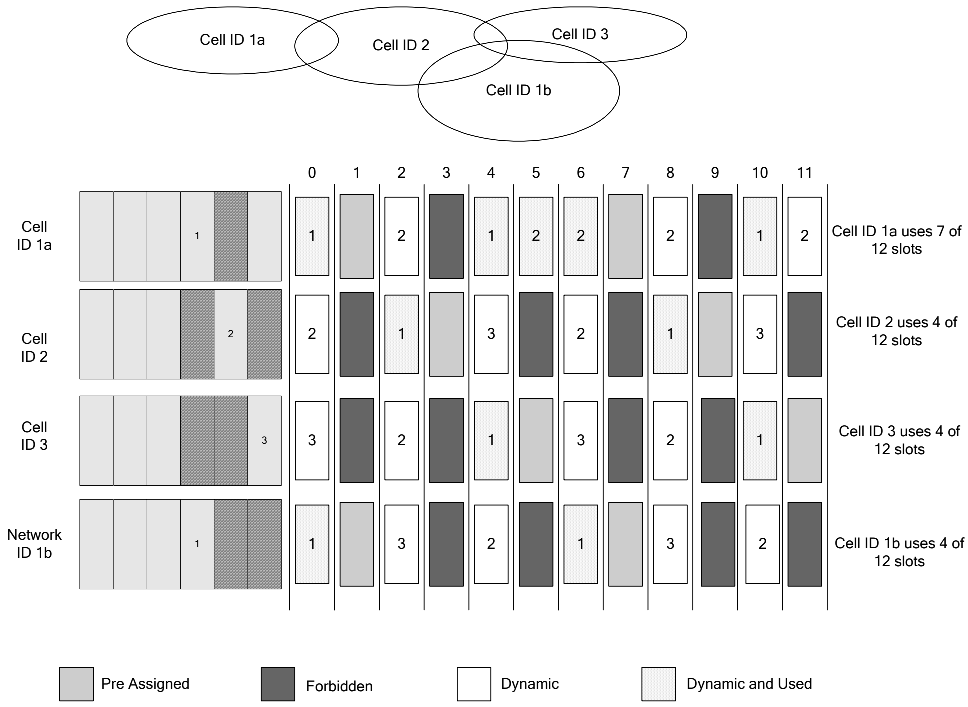

Figure 96 Reutilization with cells without total visibility

In this example, several slots have been reused. The sum of the slots used is 7+4+4+4 = 19 instead of 12, which is what happens when all cells detect each other. Table 4 shows a comparison between the requirements of fairness and the attained performance.

Cell	ID selected	# neighbors	Requirement	Obtained
1a	1	1	50% (6 of 12 slots)	58,33% (7 of 12 slots)
2	2	3	25% (3 of 12 slots)	33,33% (4 of 12 slots)
3	3	2	33,33% (4 of 12 slots)	33,33% (4 of 12 slots)
1b	1	2	33,33% (4 of 12 slots)	33,33% (4 of 12 slots)

Table 42 Performance Table for Scenario B

As it can be seen, all cells use at least the amount of transmission time that should be allocated to them, and in the case of cell 1a and cell 2, they use one more slot.

B.7.3.3 Scenario C. One Access cell and four In-home cells, all operating in FB3.

Scenario C shows how the coexistence mechanism works for scenarios where In-home and Access cells have to coexist. Figure 17 shows a scenario where there is an Access cell and four In-home cells (In-home cells do not have complete visibility). The Access cell is using the time division coexistence protocol. That is, the Access cell is using the entire FB3 band.

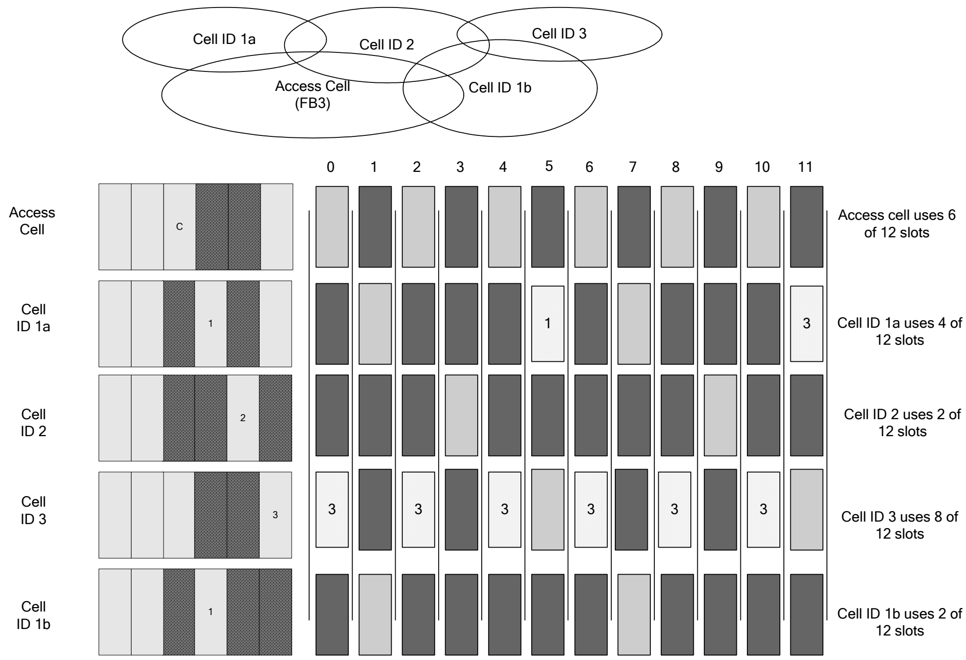

Figure 97 Access cell and several In-home cells, operating in FB3

The Access cell takes 50% of the slots (6 out of 12), and the In-home cells equally share the other 50%. The slots pre-assigned to the Access cell are the even numbered slots.

In this example, several slots have been reused. The sum of the slots used is 6+4+2+8 +2= 22 instead of 12. Table 5 depicts a comparison of the required performance versus the actual obtained performance.

Cell	ID selected	# inhome neighbors	Requirement	Obtained
Access (FB3)	B	2	50% (6 of 12 slots)	50% (6 of 12 slots)
1a	1	1 + Access Time Division	25% (3 of 12 slots)	33,33% (4 of 12 slots)

2	2	3 + Access Time Division	12.5% (1.5 of 12 slots)	16.67% (2 of 12 slots)
3	3	2	33,33% (4 of 12 slots)	66.67% (8 of 12 slots)
1b	1	2 + Access Time Division	16.67% (2 of 12 slots)	16.67% (2 of 12 slots)

Table 43 Performance Table for Scenario C

Again, it can be seen that every cell uses the required amount of slots, and in some cases, such as cells 1a, 2 and 3, the number of slots used is higher than required**.**

B.7.3.4 Scenario D. Two Access cells (one operating in FB1, the other in FB2) and four In-home cells operating in FB2.

Scenario D shows how the coexistence mechanism works for scenarios where In-home and Access cells have to coexist. The difference between scenario C and D is that two overlapped Access cells (one using FB1, the other one using FB2) are involved in scenario D The result is that indoor cells 1a, 2 and 1b will sense positions A and B of the presence window. Access cell FB2 is subject to time division rules as depicted in Figure 18 whereas Access cell FB1 is not subject to any time constraints on the data window part of the frame.

The algorithm is exactly the same as in scenario C, varying the slots reserved for Access. In this case, the Access cell operating in FB2 has slots 0, 4 and 8 reserved, instead of all even numbered slots.

Figure 18 shows how this scenario is resolved.

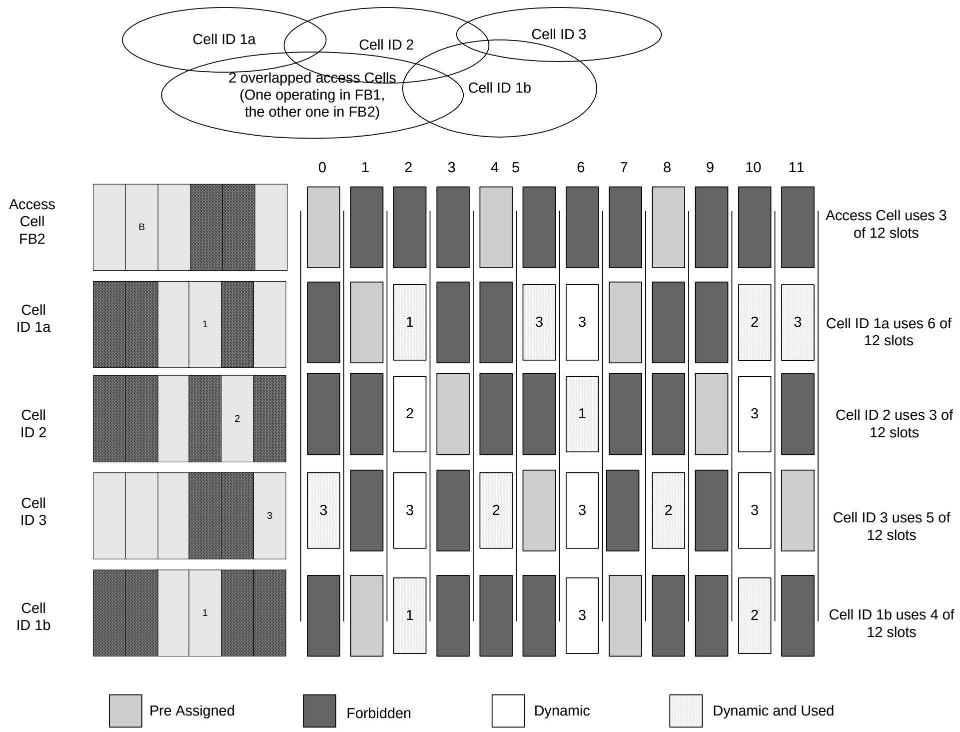

Figure 98 Overlapped Access cells operating in FB1 and FB2 with several In-home cells

The Access requirement is fulfilled: one of each 4 slots is for the Access cell operating in FB2 which is operating in conjunction with a locally detected FB1 Access cell) and the remaining slots will be shared by In-home cells. Table 6 depicts a comparison of the required performance versus the actual obtained performance.

Cell	**ID selected**	**# inhome neighbors**	**Requirement**	**Obtained**
Access cell operating in FB2	B	2	25% (3 of 12 slots)	25% (3 of 12 slots)
1a	1	1 + Access (B)	37.5% (4.5 of 12 slots)	50% (6 of 12 slots)
2	2	3 + Access (B)	18.75%	25%

			(2.25 of 12slots)	(3 of 12slots)
3	3	2	33,33% (4 of 12slots)	41,67% (5 of 12 slots)
1b	1	2 + Access (B)	25% (3 of 12 slots)	33.33% (4 of 12 slots)

Table 44 Performance Table for Scenario D

It can be seen that every cells uses more than the required amount of slots.