

## ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



### PROYECTO FIN DE CARRERA

**“Implementación software del nivel MAC del  
estándar IEEE.802.15.7 (VLC) para la plataforma  
OMNeT++”**

**Autor: Carolina Martínez Pérez  
Tutores: Itziar Alonso González**

**Fecha: Marzo 2017**



## ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



### PROYECTO FIN DE CARRERA

**“Implementación software del nivel MAC del  
estándar IEEE.802.15.7 (VLC) para la plataforma  
OMNeT++”**

### HOJA DE FIRMAS

**Alumno/a**

Fdo.: Carolina Martínez Pérez

**Tutora**

Fdo.: Itziar Alonso González

**Fecha: Marzo 2017**



## ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



### PROYECTO FIN DE CARRERA

**“Implementación software del nivel MAC del estándar  
IEEE.802.15.7 (VLC) para la plataforma OMNeT++”**

### HOJA DE EVALUACIÓN

**Calificación:** \_\_\_\_\_

**Presidente**

Fdo.:

**Vocal**

Fdo.:

**Secretario/a**

Fdo.:

**Fecha: Marzo 2017**



# ABSTRACT

---

In this report it described the development of a software package that implements the level of media access layer, IEEE 802.15.7 MAC protocol standard, to be included as a module within the simulation tool OMNET++. The MAC module developed allows the modeling of the main functions, protocols and services of the MAC level for short-range non-guided optical communications using visible light (VLC) based on the IEEE-802.15.7 standard.



# RESUMEN

---

## **Proyecto Fin de Carrera:**

**Título:** *Implementación software del nivel MAC del estándar IEEE.802.15.7 (VLC) para la plataforma OMNeT++*

**Autora:** Carolina Martínez Pérez

**Tutores:** Itziar Alonso González

En esta memoria se ha descrito el desarrollo de un paquete software que implementa el nivel de capa de acceso al medio, protocolo MAC del estándar IEEE 802.15.7, para incluirla como un módulo dentro de la herramienta de simulación OMNet++. El módulo MAC desarrollado permite modelar las principales funciones, protocolos y servicios del nivel MAC para comunicaciones ópticas no guiadas de corto alcance usando luz visible (VLC), en base a lo establecido en el estándar IEEE-802.15.7.

El objetivo principal de este proyecto final de carrera fue crear una versión del protocolo MAC que soportase los siguientes puntos:

1. Modo con baliza habilitada.
2. Mecanismo de acceso múltiple por detección de portadora con y sin evasión de colisiones (CSMA y CSMA/CA) para mecanismo de acceso con baliza ranurado.
3. Modelo de transferencia directo y GTS (ranuras de tiempo garantizadas).
4. Control de espaciado entre tramas, filtrado y detección de duplicidad.
5. Control de temporizaciones según la unidad de tiempo de relojes ópticos.
6. Implementación de las primitivas necesarias para el funcionamiento de los puntos anteriores.
7. Creación del formato de tramas para VLC, incluyendo las tramas CVD para mantener la intensidad de luz apropiada entre tramas de datos, soportar control de luminosidad, y para el suministro de información visual tales como el estado de la comunicación y la calidad del canal al usuario

**Palabras clave:** comunicación por luz visible, control de acceso al medio, OMNET++, 802.15.7.



# SUMMARY

---

## **M.S. Thesis:**

**Title:** *Software implementation of the MAC level of the IEEE.802.15.7 (VLC) standard for the OMNeT ++ platform*

**Author:** Carolina Martínez Pérez

**Advisor:** Itziar Alonso González

In this report it described the development of a software package that implements the level of media access layer, IEEE 802.15.7 MAC protocol standard, to be included as a module within the simulation tool OMNET++. The MAC module developed allows the modeling of the main functions, protocols and services of the MAC level for short-range non-guided optical communications using visible light (VLC) based on the IEEE-802.15.7 standard.

The main objective of this M.S. Thesis was to create a version of the MAC protocol that supported the following points:

1. Beacon-enabled mode.
2. Carrier sense multiple access detection mechanism with and without collision avoidance (CSMA and CSMA/CA) for slotted beacon access mechanism.
3. Direct data transfer model and GTS (guaranteed time slots).
4. Control of spacing between frames, filtering and detection of duplicity.
5. Control of timings according to the time unit of optical clocks.
6. Implementation of the primitives required for the operation of the previous points.
7. Creation of frame format for VLC, including CVD frames to maintain the appropriate light intensity between data frames, support control of brightness, and for the provision of visual information such as communication status and channel quality to user.

**Key terms:** visible light communication, medium Access control, OMNET++, 802.15.7.



# ÍNDICE

---

<b>Abstract</b>	<i>i</i>
<b>Resumen</b>	<i>iii</i>
<b>Summary</b>	<i>v</i>
<b>Índice</b>	<i>vii</i>
<b>Índice de Figuras</b>	<i>xv</i>
<b>Índice de Tablas</b>	<i>xix</i>
<b>Índice de Código</b>	<i>xxi</i>
<b>MEMORIA</b>	
<b>Capítulo 1. Introducción</b>	<i>1</i>
1.1 Introducción	<i>1</i>
1.2 Ámbito del proyecto	<i>1</i>
1.3 Motivación y objetivos	<i>3</i>
1.4 Estructura de la memoria	<i>4</i>
<b>Capítulo 2. Comunicaciones por Luz Visible: VLC</b>	<i>7</i>
2.1 Introducción	<i>7</i>
2.2 Antecedentes de las comunicaciones por luz visible	<i>7</i>
2.2.1 Evolución de las comunicaciones por luz visible	<i>10</i>
2.3 Estado del arte	<i>11</i>
2.4 De radiofrecuencia a VLC	<i>12</i>
2.5 Características de VLC	<i>13</i>
2.6 Ventajas y desventajas de VLC	<i>15</i>
2.7 Comparación de VLC con otros sistemas inalámbricos RF	<i>17</i>
2.7.1. VLC vs IrC	<i>17</i>
2.7.2. VLC vs RF	<i>19</i>
2.8 Aplicaciones de VLC	<i>21</i>
2.9 Retos de VLC	<i>24</i>
2.10 Componentes de un sistema VLC	<i>26</i>
2.11 Conclusiones	<i>28</i>

<b>Capítulo 3. El estándar IEEE802.15.7: nivel MAC</b>	<b>29</b>
3.1 Introducción	29
3.2 Organismos de normalización y estándares	29
3.3 El estándar IEEE 802.15.7	31
3.3.1 Tipos de dispositivos	31
3.3.2 Topologías de red	32
3.3.3 Arquitectura de un dispositivo IEEE 802.15.7	33
3.3.4 Modelo de transferencia de datos en IEEE 802.15.7	34
3.3.4.1 Transferencia de datos a un coordinador	35
3.3.4.2 Transmisión de datos desde un coordinador	35
3.3.4.3 Transferencia de datos punto a punto	35
3.3.5 Concepto de primitivas	36
3.3.6 Seguridad	36
3.4 La capa MAC	37
3.4.1 Estructura de la supertrama	38
3.4.2 Control de acceso al medio: acceso aleatorio	39
3.4.3 Formato general y tipos de tramas MAC	43
3.4.4 Negociación del reloj en la capa MAC	45
3.4.5 Constantes y atributos de la capa MAC	46
3.4.6 Otras funciones de la capa MAC	47
3.4.6.1 Inicio y mantenimiento de una VPAN	47
3.4.6.2 Asociación y disociación	47
3.4.6.3 Soporte de la función de color	47
3.4.6.4 Soporte para la iluminación y el control de la luminosidad	47
3.4.6.5 Movilidad	47
3.4.6.6 Espaciado entre tramas (IFS)	47
3.5. Comparación de IEEE-802.15.7 y IEEE-802.15.4	48
3.6 Conclusiones	49
<b>Capítulo 4 Herramientas de simulación: OMNeT++</b>	<b>51</b>
4.1 Introducción	51
4.2 Descripción del estado del arte en simulación de red	51
4.2.1. NS-2 (Network Simulator 2)	52
4.2.2. OPNET Modeler	52
4.2.3. GNS3	53
4.2.4. CNET Network Simulator	53
4.2.5. J-Sim	54
4.2.6. VisualSense	54
4.2.7. OMNEST	55
4.2.8. OMNeT++ (OMNeT 2008)	55

4.3 Entorno de trabajo en OMNeT++	56
4.3.1 Ficheros <i>.msg</i>	57
4.3.2 Ficheros <i>.ned</i>	58
4.3.3 Simulación con OMNeT++	60
4.3.4 Configuración de simulaciones en OMNeT++	61
4.3.4.1 Configuración de las funciones estadísticas	61
4.3.4.2 Configuración del tiempo de simulación	61
4.3.4.2 Configuración de la grabación de resultados	61
4.3.5 Análisis de resultados	62
4.4 Marco de trabajo para OMNeT++	62
4.4.1 Descripción de INET	63
4.4.2 Arquitectura de INETMANET	64
4.4.3 Módulos y protocolos	64
4.4.4 Comunicación entre capas de protocolos	66
4.5 Análisis de las funcionalidades de acceso al medio de INETMANET	67
4.6 Conclusiones	69
<b>Capítulo 5. Implementación de la capa MAC</b>	<b>70</b>
5.1 Introducción	72
5.2 Descripción funcional del modelo	72
5.3 Metodología de trabajo	72
5.4 Tareas de programación realizadas	73
5.5 Acondicionamiento del código	74
5.5.1 Estructura Task Pending	74
5.5.2 Formato trama. Cabecera MAC, MHR	78
5.5.3 Todas las tramas	79
5.5.4 Tramas de baliza	80
5.5.5 Tramas de datos	80
5.5.6 Control de trama y supertrama	80
5.5.7 Campos de información GTS	80
5.5.8 Trama CVD	81
5.5.9 Comando de información	81
5.6 Modo con baliza habilitada	81
5.6.1 Solución #1	82
5.6.2 Solución #2	82
5.6.3 Pérdida de sincronización	82
5.6.3.1 Caso #1	83
5.6.3.2 Caso #2	85
5.7 Mecanismo de acceso múltiple por detección de portadora con y sin evasión de colisiones	86
5.7.1 FSM	86

5.7.2 Algoritmo de acceso al canal	87
5.7.2.1 Propuesta 01	87
5.7.2.1.1 Cambio 01	87
5.7.2.2 Propuesta 02	88
5.7.2.2.1 Cambio 01	88
5.7.2.3 Propuesta 03	88
5.7.2.3.1 Cambio 01	88
5.7.2.3.2 Cambio 02	89
5.7.2.3.3 Cambio 03	90
5.7.2.3.4 Cambio 04	92
5.7.2.3.5 Cambio 05	95
5.8 Modo indirecto	95
5.8.1 Rutina del coordinador para una transacción en modo indirecto	95
5.8.2 Rutina del dispositivo para solicitar un dato en una transacción en modo Indirecto	96
5.8.3 Comando de solicitud de dato	97
5.8.4 Algoritmo para el comando de solicitud de dato en el dispositivo	97
5.8.5 Algoritmo de gestión para la recepción de un comando de solicitud de dato en el dispositivo	97
5.9 Conclusiones	97
<b>Capítulo 6. Simulaciones y análisis</b>	<b>99</b>
6.1 Introducción	99
6.2 Parametrización de las simulaciones	99
6.3 Parametrización de las estadísticas de resultados	101
6.4 Simulaciones de comprobación	102
6.4.1 Escenario <i>BasicEscenario2_nodos</i>	102
6.4.2 Resultados y discusión de resultados	105
6.5 Modo con baliza habilitada	106
6.5.1 Simulación de solución #2	107
6.5.2 Pérdida de sincronización	109
6.5.2.1 Simulación del Caso #1	109
6.6 Modo directo	110
6.7 Conclusiones	113
<b>Capítulo 7. Conclusiones y Líneas futuras</b>	<b>115</b>
7.1 Objetivos alcanzados	115
7.2 Líneas futuras	116
<b>8. Pliego de condiciones</b>	<b>119</b>

8.1 Objeto	119
8.2 Condiciones generales	119
8.3 Condiciones específicas	119
8.3.1 Especificaciones de equipamiento informático	120
8.3.2 Especificaciones de equipamiento informático	120
8.3.3 Instalación del paquete <i>simVLC</i>	120
8.3.4 Uso básico del simulador <i>simVLC</i>	121
<b>9. Presupuesto</b>	<b>125</b>
9.1 Objeto	125
9.2 Coste de personal	125
9.3 Coste de recursos materiales	126
9.4 Coste del proyecto	127
<b>10. Bibliografía</b>	<b>129</b>
<b>Anexo I. Especificación del servicio de la capa MAC del estándar EEE802.15.7</b>	<b>137</b>
I.1 Servicio de datos MAC	138
I.1.1 MCPS-DATA.request	138
I.1.2 MCPS-DATA.confirm	139
I.1.3 MCPS-DATA.indication	140
I.1.4 MCPS-PURGE.request	141
I.1.5 MCPS- PURGE.confirm	141
I.1.6 Secuencia de mensajes del servicio de datos	142
I.2 Servicio de gestión MAC	142
I.2.1 MLME-SAP: Primitivas de asociación	144
I.2.1.1 MLME-ASSOCIATE.request	144
I.2.1.2 MLME-ASSOCIATE.indication	144
I.2.1.3 MLME-ASSOCIATE.response	144
I.2.1.4 MLME-ASSOCIATE.confirm	144
I.2.1.5 Secuencia de mensajes de asociación	144
I.2.2 MLME-SAP: Primitivas de des-asociación	145
I.2.2.1 MLME-DISASSOCIATE.request	145
I.2.2.2 MLME-DISASSOCIATE.indication	145
I.2.2.3 MLME-DISASSOCIATE.confirm	145
I.2.2.4 Secuencia de mensajes de des-asociación	145
I.2.3 MLME-SAP: Primitiva de notificación de baliza	146
I.2.3.1 MLME- BEACON-NOTIFY.indication	146
I.2.4 MLME-SAP: Primitivas de lectura de atributos PIB	146

I.2.4.1 MLME-GET.request	146
I.2.4.2 MLME-GET.confirm	146
I.2.5 MLME-SAP: Primitivas de gestión GTS	146
I.2.5.1 MLME-GTS.request	146
I.2.5.2 MLME-GTS.indication	146
I.2.5.3 MLME-GTS.confirm	146
I.2.5.4 Secuencia de mensajes de gestión GTS	147
I.2.6 MLME-SAP: Primitivas para el reinicio de la subcapa MAC	148
I.2.6.1 MLME-RESET.request	148
I.2.6.2 MLME-RESET.confirm	148
I.2.7 MLME-SAP: Primitivas para la especificación del tiempo de habilitación del receptor	148
I.2.7.1 MLME-RX-ENABLE.request	148
I.2.7.2 MLME-RX-ENABLE.confirm	148
I.2.8 MLME-SAP: Primitivas para el escaneo de canales	148
I.2.8.1 MLME-SCAN.request	148
I.2.8.2 MLME-SCAN.confirm	149
I.2.8.3 MLME-RX-ENABLE.confirm	149
I.2.9 MLME-SAP: Primitivas del estado de comunicación	149
I.2.9.1 MLME-COMM-STATUS.indication	149
I.2.10 MLME-SAP: Primitivas para la escritura de los atributos PIB	149
I.2.10.1 MLME-SET.request	149
I.2.10.2 MLME-SET.confirm	149
I.2.11 MLME-SAP: Primitivas para para la actualización de la configuración de supertrama	149
I.2.11.1 MLME-START.request	150
I.2.11.2 MLME-START.confirm	150
I.2.11.3 Secuencia de mensajes para la actualización de la configuración de supertrama	150
I.2.12 MLME-SAP: Primitivas para para la sincronización con un coordinador	150
I.2.12.1 MLME-SYNC.request	150
I.2.13 MLME-SAP: Primitivas para la pérdida de sincronización con un coordinador	150
I.2.13.1 MLME-SYNC-LOSS.indication	150
I.2.13.2 Secuencia de mensajes para la sincronización con un coordinador	151
I.2.14 MLME-SAP: Primitivas para la petición de datos de un coordinador	152
I.2.14.1 MLME-POLL.request	152
I.2.14.2 MLME-POLL.confirm	152
I.2.14.3 Secuencia de mensajes para la petición de datos de un coordinador	152
I.2.15 Primitivas PD-SAP	153

I.2.15.1 PD-DATA.request	153
I.2.16 Primitivas PLME-SAP	153
<b>Anexo II. FSM para la transmisión de trama de datos con CSMA/CA</b>	<b>155</b>
II.1 Variables implicadas en el control de la FSM	155
II.1.1 Task	155
II.1.2 TaskP	155
II.1.3 frmCtrl	156
II.1.4 tmpData	157
II.1.5 txData	157
II.1.6 txCsmaca	157
II.1.7 tmpCsmaca	157
II.1.8 waitDataAck	157
II.1.9 numDataRetry	157
II.1.10 NB	157
II.1.11 BE	157
II.1.12 CW	157
II.1.13 backoffStatus	158
II.1.14 Otras	158
II.2 Métodos implicados en el control de la FSM	158
II.2.1 MCPS_DATA_request	159
II.2.2 checkTaskOverflow	159
II.2.3 dispatch	159
II.2.4 FSM_MCPS_DATA_request	160
II.2.5 taskFailed	161
II.2.6 taskSuccess	161
II.3 Diagrama de estados de la FSM	161
<b>Anexo III. Algoritmo de acceso al canal de la norma IEEE.802.15.7</b>	<b>163</b>
III.1 Algoritmo de acceso aleatorio	163
III.2 Diferencias con el algoritmo de acceso al canal de la norma IEEE.802.15.4	166
III.3 Implementación en WPAN de INETMANET del CSMA-CA del IEEE.802.15.4	167
III.3.1 Métodos y funciones implicadas	168
<b>Anexo IV. Gestión de transacción. Modo indirecto</b>	<b>173</b>
IV.1 Gestión de transacción	173
IV.2 Extracción de un dato pendiente del coordinador	174



# ÍNDICE DE FIGURAS

---

<b>Capítulo 2. Comunicaciones por Luz Visible: VLC</b>	<b>7</b>
Figura 2.1 a) Baliza s. XVI (Devon, UK). b) Heliógrafo Mance (1910). c) Modulación en el transmisor de un fotófono.	9
Figura 2.2 Espectro radioeléctrico.	10
Figura 2.3 VLC para aplicaciones de localización en interiores.	24
Figura 2.4 Diferentes aplicaciones de VLC: comunicaciones a bordo de avión, en oficina/vivienda, en hospitales, en automoción y tráfico, entretenimiento y realidad aumentada.	24
Figura 2.5 Esquema de un sistema VLC básico.	27
Figura 2.6 Esquema ejemplo de un sistema VLC básico con tecnología OFDM.	28
<b>Capítulo 3. El estándar IEEE802.15.7: nivel MAC</b>	<b>29</b>
Figura 3.1 Topologías de red en el estándar IEEE 802.15.7.	33
Figura 3.2 Arquitectura de un dispositivo IEEE 802.15.7.	34
Figura 3.3 Concepto de primitiva de servicio.	36
Figura 3.4 Estructura de supertrama MAC.	39
Figura 3.5 Versión ranurada del algoritmo de acceso aleatorio.	40
Figura 3.6 Diagrama de flujo del mecanismo de acceso aleatorio para todos los accesos.	41
Figura 3.7 Formato general de las tramas MAC.	44
Figura 3.8 Negociación de reloj en la capa MAC.	45
<b>Capítulo 4 Herramientas de simulación: OMNeT++</b>	<b>51</b>
Figura 4.1 Entorno gráfico <i>Tkenv</i> .	56
Figura 4.2 Diseño fichero <i>.ned</i>	60
Figura 4.3 (a) Estructura de directorios INETMANET, vs. (b) Estructura de directorios VLC.	65
Figura 4.4 (a) Estructura conceptual del modelo IEEE.802.15.7. (b) Estructura implementada de la capa de enlace del modelo IEEE.802.15.7.	65
<b>Capítulo 5. Implementación de la capa MAC</b>	<b>70</b>
Figura 5.1 Módulos componentes de cada nodo de la red.	72
Figura 5.2 Eliminación de SIZE_OF_802157_ORPHAN_NOTIFICATION.	76
Figura 5.3 Eliminación del tipo ORPHAN_SCAN en tipo de exploración.	76
Figura 5.4 Eliminación del tipo de comando Ieee802157_ORPHAN_NOTIFICATION.	77
Figura 5.5 Eliminación del tipo de tarea TP_MLME_ORPHAN_RESPONSE.	77
Figura 5.6 Eliminación del comando Ieee802157OrphanNotiCmd.	77
Figura 5.7 Eliminación del mensaje Ieee802157OrphanNotiCmd.	78

<b>Figura 5.8</b>	Método “ <i>handleCommand</i> ”.	<b>78</b>
<b>Figura 5.9</b>	Método “ <i>calFrmByteLength</i> ”.	<b>78</b>
<b>Figura 5.10</b>	Eliminación del campo <i>intraPan</i> en tipo de trama.	<b>78</b>
<b>Figura 5.11</b>	Método “ <i>constructACK</i> ”.	<b>79</b>
<b>Figura 5.12</b>	Método “ <i>MCPS_DATA_request</i> ”.	<b>79</b>
<b>Figura 5.13</b>	Método “ <i>handleBcnTxTimer</i> ”.	<b>79</b>
<b>Figura 5.14</b>	Cambio 01 de la propuesta 01: contabilizar los reintentos de Backoff también dentro de la FSM.	<b>87</b>
<b>Figura 5.15</b>	Cambio 01 de la propuesta 02: llamada a “ <i>csmacaReset</i> ” en el método “ <i>csmacaEntry</i> ”.	<b>88</b>
<b>Figura 5.16</b>	Cambio 02 de la propuesta 02: eliminar la llamada a “ <i>csmacaReset</i> ” en el método “ <i>csmacaStart</i> ”.	<b>88</b>
<b>Capítulo 6.</b>	<b>Simulaciones y análisis</b>	<b>99</b>
<b>Figura 6.1</b>	Transmisión de baliza de supertrama.	<b>103</b>
<b>Figura 6.2</b>	Transmisión de datos en modo directo.	<b>103</b>
<b>Figura 6.3</b>	Supertrama	<b>105</b>
<b>Figura 6.4</b>	Eventos <i>txSDTimer</i> y <i>bcnTxTimer</i> en la simulación inicial.	<b>106</b>
<b>Figura 6.5</b>	Comportamiento de la capa PHY tras recibir trama <i>Ieee802157BEACON</i> de transmisión.	<b>107</b>
<b>Figura 6.6</b>	Resultado de simulación con solución #2. Eventos <i>txSDTimer</i> y <i>bcnTxTimer</i> en la simulación final.	<b>108</b>
<b>Figura 6.7</b>	Resultado de simulación con solución #2. Detalle del parámetro “ <i>goodput</i> ”.	<b>108</b>
<b>Figura 6.8</b>	Resultado de simulación con solución #2. Detalle del parámetro “ <i>Mean end-to-end delay</i> ”.	<b>109</b>
<b>Figura 6.9</b>	Instante real de recepción del beacon (1,048576003332).	<b>110</b>
<b>Figura 6.10</b>	Instante programado para la recepción del beacon (1,048576003333).	<b>110</b>
<b>Figura 6.11</b>	Configuración de una prueba del modo indirecto iniciada por coordinador.	<b>111</b>
<b>Figura 6.12</b>	Resultado de la simulación del modo indirecto iniciada por coordinador.	<b>111</b>
<b>Figura 6.13</b>	Configuración de una prueba del modo indirecto iniciada por coordinador: caso lista llena.	<b>112</b>
<b>Figura 6.14</b>	Resultado de la simulación del modo indirecto iniciada por coordinador: caso lista llena.	<b>112</b>
<b>8.</b>	<b>Pliego de condiciones</b>	<b>119</b>
<b>Figura 8.1</b>	Cuadros de diálogos para la importación de proyectos existentes a un espacio de trabajo de OMNeT++.	<b>121</b>
<b>Figura 8.2</b>	Botón de inicio de simulaciones, con diferentes escenarios configurados en el lanzador.	<b>122</b>
<b>Figura 8.3</b>	Captura de fichero de configuración de parámetros NED abierto.	<b>123</b>

<b>Anexo I. Primitivas nivel MAC del estándar IEEE802.15.7</b>	<b>137</b>
<b>Figura I.1</b> Modelo de referencia para la subcapa MAC.	<b>137</b>
<b>Figura I.2</b> Secuencia de mensajes describiendo el servicio de datos MAC.	<b>138</b>
<b>Figura I.3</b> Secuencia de mensajes para la asociación.	<b>142</b>
<b>Figura I.4</b> Secuencia de mensajes para la des-asociación iniciada por un dispositivo.	<b>144</b>
<b>Figura I.5</b> Secuencia de mensajes para la asignación de un GTS iniciada por un dispositivo.	<b>145</b>
<b>Figura I.6</b> Secuencia de mensajes para el rehúso iniciado por un dispositivo (a) y el coordinador PAN (b).	<b>147</b>
<b>Figura I.7</b> Secuencia de mensajes para la actualización de la configuración de supertrama.	<b>147</b>
<b>Figura I.8</b> Secuencia de mensajes para la sincronización con un coordinador en una VPAN con uso de baliza.	<b>151</b>
<b>Figura I.9</b> Secuencia de mensajes para la petición de datos desde un coordinador.	<b>151</b>
<b>Anexo II. FSM para la transmisión de trama de datos con CSMA/CA</b>	<b>155</b>
<b>Figura II.1</b> Diagrama de estados de la FSM.	<b>162</b>
<b>Anexo III. Primitivas nivel MAC del estándar IEEE802.15.7</b>	<b>163</b>
<b>Figura III.1</b> Diagrama de flujo del acceso aleatorio.	<b>164</b>
<b>Figura III.2</b> Diagrama de flujo del CSMA-CA.	<b>165</b>
<b>Figura III.3</b> Comparación de algoritmos.	<b>166</b>



# ÍNDICE DE TABLAS

---

<b>Capítulo 2. Comunicaciones por Luz Visible: VLC</b>	<b>7</b>
<b>Tabla 2.1</b> Ventajas y desventajas de VLC.	<b>16</b>
<b>Tabla 2.2</b> Comparación entre sistemas VLC e IrC.	<b>18</b>
<b>Tabla 2.3</b> Comparación entre sistemas VLC e RF.	<b>20</b>
<b>Capítulo 3. El estándar IEEE802.15.7: nivel MAC</b>	<b>29</b>
<b>Tabla 3.1</b> Tipos de dispositivos y sus características (IEEE 802.15.7).	<b>32</b>
<b>Tabla 3.2</b> Principales constantes de la capa MAC (IEEE 802.15.7).	<b>46</b>
<b>Tabla 3.3</b> Principales atributos de la capa MAC (IEEE 802.15.7).	<b>46</b>
<b>Capítulo 4 Herramientas de simulación: OMNeT++</b>	<b>51</b>
<b>Tabla 4.1</b> Módulo simple <i>NotificationBoard</i> .	<b>66</b>
<b>Tabla 4.2</b> Módulo simple <i>ChannelControl</i>	<b>67</b>
<b>Capítulo 5. Implementación de la capa MAC</b>	<b>70</b>
<b>Tabla 5.1</b> Formato del campo de carga útil de trama de datos.	<b>74</b>
<b>Tabla 5.2</b> Formato del campo de carga útil de trama de datos.	<b>80</b>
<b>Capítulo 6. Simulaciones y análisis</b>	<b>99</b>
<b>Tabla 6.1</b> Formato del campo de carga útil de trama de datos.	<b>100</b>
<b>Tabla 6.2</b> Formato del campo de carga útil de trama de datos.	<b>100</b>
<b>Tabla 6.3</b> Módulos para la topología en estrella de red VLC, <i>Ieee802157StarNet</i> .	<b>101</b>
<b>Tabla 6.4</b> Parámetros principales para la configuración de escenarios.	<b>101</b>
<b>Tabla 6.5</b> Variables de sistema para recopilación de estadísticas.	<b>102</b>
<b>Tabla 6.6</b> Configuración característica inicial de los parámetros de definición de supertrama.	<b>104</b>
<b>Tabla 6.7</b> Configuración característica inicial de los parámetros de definición de supertrama.	<b>104</b>
<b>Tabla 6.8</b> Valores iniciales de super-trama de simulación.	<b>104</b>
<b>9. Presupuesto</b>	<b>125</b>
<b>Tabla 9.1</b> Aproximación de la dedicación temporal en las distintas fases del desarrollo de la librería MAC del IEEE 802.15.7.	<b>125</b>
<b>Tabla 9.2</b> Costes por dedicación temporal de personal, o mano de obra.	<b>126</b>
<b>Tabla 9.3</b> Desglose de los costes por recursos materiales.	<b>126</b>

<b>Anexo I. Primitivas nivel MAC del estándar IEEE802.15.7</b>	<b><i>137</i></b>
<b>Tabla I.1 Primitivas MCPS-SAP</b>	<b><i>138</i></b>
<b>Tabla I.2 Primitivas MLME-SAP</b>	<b><i>143</i></b>
<b>Tabla I.3 Primitivas MLME-SAP</b>	<b><i>143</i></b>
<b>Tabla I.4 Primitivas PD-SAP</b>	<b><i>143</i></b>
<b>Tabla I.5 Primitivas PMLE-SAP</b>	<b><i>143</i></b>

# ÍNDICE DE CÓDIGO

---

<b>Capítulo 4</b> Herramientas de simulación: <i>OMNeT++</i>	<b>51</b>
Código 4.1 Ejemplo de estructura de mensaje <i>Command Frame</i> .	<b>57</b>
Código 4.2 Ejemplo de código de cabecera generado automáticamente para <i>Command Frame</i>	<b>58</b>
Código 4.3 Código fuente del fichero <i>.ned</i> . correspondiente al módulo de la <b>Figura 4.2</b> .	<b>59</b>
<b>Capítulo 5.</b> Implementación de la capa MAC	<b>70</b>
Código 5.1 Código inicial del método <i>cmacaReset</i> .	<b>89</b>
Código 5.2 Código para <i>Propuesta 03. Cambio 02</i> .	<b>90</b>
Código 5.3 Código para <i>Propuesta 03. Cambio 03</i> .	<b>91</b>
Código 5.4 Código para <i>Propuesta 03. Cambio 03</i> (continuación).	<b>92</b>
Código 5.5 Código para <i>Propuesta 03. Cambio 04</i> .	<b>93</b>
Código 5.6 Código para <i>Propuesta 03. Cambio 04</i> (continuación).	<b>94</b>



# Capítulo 1

## Introducción

---

*"No es que no nos atrevamos porque las cosas sean difíciles, simplemente las hacemos difíciles cuando no nos atrevemos"*

LUCIO SÉNECA (4a.C., 65d.C.)

### 1.1. Introducción

Este capítulo recoge el planteamiento del presente proyecto. Se describe el ámbito de trabajo, así como los antecedentes que enmarcan el desarrollo a realizar. Se presentan los objetivos propuestos y alcanzados, las herramientas con las que se ha contado y la organización de la memoria del proyecto.

### 1.2. Ámbito del proyecto

Los considerables cambios e innovaciones que tienen lugar en la transmisión de datos de alta velocidad en las últimas décadas, han dado lugar a la aparición de una gran variedad de sistemas de comunicación. Debido a esta diversificación de sistemas de transmisión de datos, surgen continuamente nuevas especificaciones según el ámbito de las redes, sus necesidades, y finalidad, cuya eficiencia va mejorando en relación a las limitaciones físicas de cada entorno, seguridad, capacidad del medio, y otras características.

En particular, las redes de comunicación inalámbrica han adquirido una suma importancia en los últimos tiempos, dando lugar al uso de distintas tecnologías y a múltiples protocolos de red según la frecuencia de transmisión, velocidad y alcance. Hoy en día, el asentamiento de los dispositivos móviles en el mercado de consumo digital ha supuesto un gran empuje al desarrollo de redes inalámbricas de área local y personal. Esta demanda además plantea otros retos a la tecnología inalámbrica dados por factores como la exigencia en la solución de problemas de ámbitos domésticos e industriales.

Recientemente, tanto el desarrollo en las comunicaciones ópticas, como los avances en iluminación LED (*Light Emission Diode*), han permitido crear sistemas competitivos de comunicación por luz visible, VLC. Estos sistemas presentan ventajas frente a los sistemas basados en transmisión por radiofrecuencia tradicional, mejorando el tamaño y el costo de los transceptores, y eliminando barreras de licencia de operación, así como la posibilidad de interferencias electromagnéticas. Esta tecnología, que se encuadra en las consideradas de óptica en el espacio libre (FSO), emplea radiocomunicación por ondas electromagnéticas de espacio visible.

La tecnología VLC (*Visual Light Communication*), que data de comienzos del siglo XXI, permite transmitir datos a través de la iluminación visible enviando datos a través de luces LED que varían su intensidad a una velocidad que el ojo humano no pueda percibir. Uno de los elementos de esta tecnología es la nueva generación de LEDs de

elevado brillo. Es posible codificar datos en la luz variando la tasa a la que el LED parpadea de ON a OFF para dar diferentes cadenas de unos y de ceros binarios. La intensidad del LED se modula tan rápido que el ojo humano no lo percibe. VLC permite establecer canales dedicados y comunicaciones simplex, half-duplex y full-duplex [1,2].

Emplea diodos LED productores de luz visible se encuentran en cualquier lugar, lo que presenta una ventaja importante respecto a otros sistemas. Sus posibles usos están en desarrollo, y son complemento o alternativa de comunicación inalámbrica en hogares y oficinas, redes de respaldo, intercomunicación entre redes LAN, comunicaciones espaciales entre satélites, comunicaciones en y entre circuitos integrados, o comunicaciones inteligentes en respuesta al entorno, entre muchos otros usos. [1,2].

La principal aportación de VLC consiste en la posibilidad de unir las funciones de iluminación y comunicaciones en un mismo dispositivo, con el consecuente ahorro de recursos energéticos. Según el *Expert Advisory Group and Steering Board Members of the NetWorks Technology Platform* “la tecnología VLC es una de las ocho nuevas tecnologías donde se enfocará la investigación en los próximos años”. VLC fue para la publicación TIME Magazine uno de los cincuenta mejores inventos del año 2011 [1].

La irrupción de una nueva tecnología de comunicaciones requiere la necesidad de establecer estándares para su regulación y desarrollo, y a los cuales deben obedecer los fabricantes de dispositivos que la implementen. Son varias las organizaciones estandarizando las comunicaciones ópticas no guiadas con luz visible [3,4], pero la proporcionada por el IEEE se ha convertido en referente para VLC.

En Diciembre del 2011 el comité del *Institute of Electrical and Electronics Engineers* (IEEE) dedicado a redes de área personal publica el estándar IEEE.802.15.7, “*Shortrange Wireless Optical Communication Using Visible Light*” [5], que recoge las normas de comunicación óptica en el espacio libre usando la luz visible y define la capa física (PHY) y la capa de acceso al medio (MAC), incluyendo, para ambos niveles, las especificaciones de protocolo, comunicación entre subcapas, servicio y seguridad.

El nivel de capa de enlace o MAC (*Medium Access Control*), regula las comunicaciones a nivel de trama de datos en el medio, definiendo las reglas en que los nodos lo comparten. El estándar IEEE802.15.7, permite una estructura de supertrama opcional y define cuatro combinaciones posibles de métodos de acceso, según sea aleatorio o bajo el protocolo CSMA/CA (*Carrier Sense Multiple Access/Collision Avoidance*), ranurado o no ranurado. Lo que conlleva a poner en estudio los modelos de transferencia de datos, selección de frecuencia de reloj, y seguridad, según el método y la topología escogida para conformar una VPAN (*Visible-light communication Personal Area Network*).

VLC es una tecnología relativamente nueva que aún presenta necesidades de desarrollo, experimentación y análisis. Cualquier herramienta en este sentido sería muy útil. Entre ellas están, claramente, las herramientas de simulación por ordenador. Es usual para el modelado y estudio de las redes usar herramientas de simulación, que permitan representar escenarios y situaciones reales, así como reproducir el envío y recepción de tramas de datos entre dispositivos, conexión y desconexión, colisiones posibles problemas, distintos protocolos, etc.

El estándar IEEE 802.15.7, aunque ya no reciente, cuenta con escasos proveedores de soluciones software que lo implementen o incluyan como una componente estándar dentro de sus herramientas de simulación de red. En la mayoría de los casos, los módulos disponibles en las herramientas software que implementan la norma IEEE

802.15.7/VLC son fruto de contribuciones externas de carácter *Open Source* y, casi siempre, se encuentran en fase de desarrollo sin proporcionar demasiadas garantías de un correcto funcionamiento. Como consecuencia de este grado de inmadurez, es escasa la documentación y el soporte técnico correspondiente a dichos módulos y así el desarrollador se encuentra con la necesidad de crear sus propios módulos o adaptar los ya existentes.

Entre las alternativas vigentes, se encuentra el simulador abierto OMNeT++ [6], que ofrece una serie de ventajas frente a otros. Está compuesto de un entorno de desarrollo basado en Eclipse, con lenguaje de base C++ y un lenguaje propio de red denominado NED, un entorno de ejecución gráfica y una serie de herramientas adicionales. Al ser modular y extensible, permite la incorporación de forma sencilla de librerías de simulaciones y marcos de trabajo adicionales, desarrollados como proyectos independientes. Además, OMNeT++ es libre para uso académico, sin fines de lucro, ampliamente utilizada en la comunidad científica mundial y en continuo desarrollo.

Actualmente, el marco de trabajo INET proporciona un paquete de simulación de redes de comunicación de código abierto para el entorno de simulación OMNeT++ [7]. Contiene modelos para varios protocolos de redes cableadas e inalámbricas, como UDP, TCP, SCTP, IP, IPv6, Ethernet, PPP, 802.11, MPLS, OSPF, y muchos otros.

En particular, se ha desarrollado un proyecto INETMANET que contiene un paquete para simulación del estándar de red IEEE.802.15.4 “*Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks*” [8], punto de partida de este proyecto, ya que ambos niveles PHY y MAC tienen bastante en común con el estándar IEEE.802.15.7.

### 1.3. Motivación y objetivos

Actualmente, este proyecto fin de carrera cuenta con dos trabajos anteriores relacionados con el estándar VLC, en concreto el PFC “*Implementación software del nivel físico del estándar IEEE.802.15.7 (VLC) para la plataforma OMNeT++*” [9] y la Tesis Doctoral “*Análisis de los efectos del problema del nodo oculto en las redes IEEE 802.15.7. Propuesta de solución*” [10]. El objetivo principal de este proyecto, es seguir esta línea, pasando a implementar el siguiente nivel de capa de acceso al medio, protocolo MAC, con la finalidad de modelar redes VLC en la herramienta OMNeT++, y la simulación y estudio de prestaciones de dichos sistemas.

El objetivo principal de este proyecto es desarrollar e implementar las librerías necesarias que implementen el funcionamiento del nivel MAC del estándar IEEE802.15.7 para incluirlas en el simulador OMNET, completando así los trabajos desarrollado previamente a este PFC antes citados [9,10]. El módulo MAC deberá modelar las principales funciones, protocolos y servicios de nivel MAC para comunicaciones ópticas no guiadas de corto alcance usando luz visible, en base a lo establecido en el estándar IEEE.802.15.7.

Los objetivos a conseguir en este proyecto final de carrera es crear una versión del protocolo MAC que soporte los siguientes puntos:

1. Modo con baliza habilitada.
2. Mecanismo de acceso múltiple por detección de portadora con y sin evasión de colisiones (CSMA y CSMA/CA) para mecanismo de acceso con baliza ranurado.
3. Modelo de transferencia directo y GTS (ranuras de tiempo garantizadas).

4. Control de espaciado entre tramas, filtrado y detección de duplicidad.
5. Control de temporizaciones según la unidad de tiempo de relojes ópticos.
6. Implementación de las primitivas necesarias para el funcionamiento de los puntos anteriores.
7. Creación del formato de tramas para VLC, incluyendo las tramas CVD para mantener la intensidad de luz apropiada entre tramas de datos, soportar control de luminosidad, y para el suministro de información visual tales como el estado de la comunicación y la calidad del canal al usuario

En las comunicaciones ópticas no guiadas con luz visible, el campo de visión es limitado, por lo que los mecanismos especificados en el estándar pueden ser insuficientes para un completo control de evitar que se produzcan colisiones, es el llamado problema del nodo oculto. Por ese motivo, este proyecto final de carrera también pretende dar el soporte para la simulación del procedimiento de acceso al medio con contención basado en CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*), especificación del estándar IEEE 802.15.7.

Aunque existe la posibilidad de configurar la red con las topologías de punto a punto, estrella o difusión, se restringirá para representación de redes de topología en estrella. Además, se hará uso del marco de trabajo existente en el proyecto INETMANET sobre el protocolo para WPAN de baja velocidad, definido en el estándar IEEE 802.15.4, como base para el desarrollo del código.

Aunque el objetivo principal es añadir las funcionalidades y procedimientos de la capa MAC a los trabajos realizados anteriores sobre la capa física, [9,10] también ha sido necesario realizar los trabajos de integración con lo desarrollado para la capa física. Para ello, fue necesario modificar elementos de otras capas, introducir parametrizaciones, incluir nuevos módulos o modificar los existentes.

El conjunto de trabajo realizado permite disponer de las librerías necesarias, funciones, protocolos y servicios de la capa física, y capa MAC respecto al estándar IEEE 802.15.7. Esto ha permitido diseñar variedad de simulaciones con escenarios y condiciones concretas para estudiar, analizar y comprobar el funcionamiento de la capa MAC y los mecanismos de contienda de acceso de las redes de comunicaciones ópticas de luz visible.

Fue necesario modificar el método de configuración de las simulaciones basado en una serie de parámetros de entrada a los que se pueden asignar valores. La parametrización establecida permite configurar diferentes escenarios y condiciones a través de un amplio conjunto de parámetros que el usuario pueda establecer con facilidad de cara a sus simulaciones. Esta filosofía de interfaz de configuración parametrizable entronca con la propia del entorno de simulaciones OMNET++ sobre el cual se ha trabajado.

Finalmente se ha logrado completar el sistema de recogida de los resultados de simulación. En este aspecto el entorno de simulaciones OMNET++ empleado ofrece gran cantidad de herramientas que ayudan a la recolección de resultados, elaboración de estadísticas y posterior organización de datos y representación gráfica de los mismos.

### **1.4. Estructura de la memoria**

La realización de este proyecto se ha dividido en diferentes fases que se detallan a continuación.

El **Capítulo 1** recoge el planteamiento del presente proyecto. Se describe el ámbito de trabajo, así como los antecedentes que enmarcan el trabajo a realizar. Se presentan los objetivos propuestos y la organización de la memoria del proyecto.

El **Capítulo 2** introduce el concepto de comunicaciones por luz visible (VLC), para así acercarnos al concepto tratado en este proyecto. Describiremos en primer lugar, los fundamentos y el estado del arte de la tecnología VLC, sus orígenes, sus características principales y, por último, se enumeran sus aplicaciones. También se exponen los principales retos a afrontar en su desarrollo e implantación.

En el **Capítulo 3** se describen las principales características del estándar IEEE 802.15.7 para redes inalámbricas de área personal, WPAN, basadas en la tecnología VLC, redes que son denominadas Visible-light Communication Personal Area Network (VPAN).

El estudio del estándar IEEE.802.15.7 para VLC, es el paso previo para la implementación de una configuración de red del protocolo. En este capítulo nos centramos en el nivel MAC de dicho estándar. La subcapa MAC realiza funciones de control y gestión de tramas de los distintos protocolos, incluyendo salidas hacia la subcapa física PHY.

En el **Capítulo 4** se presentan las diversas opciones existentes en el contexto de herramientas de simulación de red que permitirían llevar a cabo la implementación de la capa MAC del estándar IEEE 802.15.7. En primer lugar, como estudio previo al desarrollo del objetivo proyecto, se ha llevado a cabo un estudio de posibles herramientas de simulación para analizar aquellas opciones que pudieran ser consideradas. Se repasan las principales características de dichas herramientas de simulación de red indicando su validez o no para la realización del objetivo de este proyecto fin de carrera.

Entre las alternativas vigentes, se encuentra el simulador abierto OMNeT++, que ofrece una serie de ventajas frente a otros. Así, a continuación del estudio del arte, se introduce la plataforma de simulación OMNeT++ y el marco de trabajo INETMAMET escogidos finalmente para implementar la capa MAC de la norma IEEE 802.15.7. Posteriormente, profundizaremos en las propiedades que se adecuan a las características del entorno de trabajo donde se desarrolla el proyecto. Es importante resaltar el hecho de que los trabajos anteriores [9,10] estuviesen hechos bajo OMNET era ya un condicionante para la elección del simulador OMNeT++. Pese a ello, se valoraron otras herramientas por si resultaba más factible implementar la capa PHY y MAC.

**Capítulo 5** describe la metodología de desarrollo empleada durante el desarrollo del módulo MAC. Se realiza una descripción de la estructura global del simulador, de sus módulos constituyentes principales y del funcionamiento de cada uno de ellos. Se presta mayor dedicación y detalle en el módulo referente a la capa MAC, que es el objetivo principal del proyecto y donde se deben lograr implementar las especificaciones del estándar.

En el **Capítulo 6** se parametrizan las simulaciones. Se simulan escenarios y condiciones concretas para comprobar el funcionamiento de los mecanismos implementados de sistemas VLC, y más concretamente, de las características de la capa MAC. El capítulo recoge la configuración de esas simulaciones, su ejecución, tratamiento de los problemas encontrados y los resultados obtenidos.

Las conclusiones y líneas futuras se recogen en el **Capítulo 7**. Se realiza un resumen de las aportaciones del trabajo realizado y de los objetivos logrados en este proyecto.

La memoria se completa con el pliego de condiciones generales y específicas para el uso del paquete de simulaciones *simVLC* implementado, incluyendo también una guía básica de uso del mismo para realizar simulaciones. Seguidamente se incorpora el presupuesto, con cálculos del tiempo empleado y una estimación del coste en el desarrollo llevado a cabo en este PFC.

La memoria finaliza con el listado de la bibliografía empleada.

También se incluyen varios anexos. En el **Anexo I** se proporciona en detalle el conjunto de primitivas de la capa MAC según el estándar IEEE802.15.7. En el **Anexo II**, **Anexo III**, y **Anexo IV** se describen detalles del protocolo de acceso al medio del estándar IEEE802.15.7 que han sido necesarios analizar para su implementación en la librería MAC objeto de este PFC.

# Capítulo 2

## Comunicaciones por Luz Visible: VLC

---

*La oscuridad no existe, la oscuridad es en realidad ausencia de luz.*

ALBERT EINSTEIN (1879-1955)

### 2.1. Introducción

Según un informe de la empresa Cisco VNI del 2011, la demanda del usuario medida en exabytes/mes está creciendo más rápidamente de lo que se gana en eficiencia espectral [1]. Así, mientras se preveía que la demanda del usuario crecería en un 32% en el período de tiempo del 2010 al 2015, la eficiencia espectral sólo podría incrementarse un 12% en el mismo período de tiempo. Cisco señala que la tasa de crecimiento anual (*Compound Annual Growth Rate*, CAGR) del uso de los datos móviles está en torno al 80%, de tal forma que ya desde el año 2015 el tráfico de los dispositivos inalámbricos supera al tráfico de los dispositivos con conexión cableada. De entre los posibles tipos de tráfico de datos, el tráfico originado en video móvil se espera que se convierta en el tráfico principal de Internet en los años venideros.

Estos incrementos en el tráfico de red exigen cambios radicales en la forma de concebir las comunicaciones inalámbricas. La tecnología VLC se presenta como una alternativa a tener en cuenta frente a las tecnologías de radiofrecuencia (RF) [1]. De entre todas las posibles, las que más desarrollo han experimentado por su modularidad y alta capacidad han sido las correspondientes a la familia de estándares 802.11 popularmente conocidos por las siglas Wi-Fi para despliegue de redes inalámbricas WLAN (IEEE802.11a/b/g/n/x1/x10/x200), y las tecnologías de telefonía móvil (GSM, GPRS, WCDMA, HSDPA/HSUPA, HSPA, LTE, 4G, 5G).

En este capítulo se va a introducir el concepto de comunicaciones por luz visible (VLC), para así acercarnos al concepto tratado en este proyecto. Describiremos en primer lugar, los fundamentos y el estado del arte de la tecnología VLC, sus orígenes, sus características principales y, por último, se enumeran sus aplicaciones. También se exponen los principales retos a afrontar en su desarrollo e implantación.

### 2.2. Antecedentes de las comunicaciones por luz visible

En la antigüedad, las señales de luz mediante hogueras o grandes faros, fueron los antiguos precursores de las comunicaciones por luz visible, y de las comunicaciones de larga distancia en general.

Una de las primeras tecnologías donde se implantó FSO fue una aplicación militar, utilizada en el ejército francés a finales del siglo XVII. El sistema de telégrafía de Chappe consistía en estructuras de madera de cinco metros de altura separados uno de otros por once kilómetros. Estaba compuesto por tres brazos movibles capaces de articular 196 señales que se traducían por palabras o frases con un significado concreto. A estas estructuras se les incorporaban telescopios para que así, desde la estructura adyacente, se pudiese ver el mensaje transmitido y enviarlo a la siguiente. En un

minuto, una señal se desplazaba 135 kilómetros. Este sistema también podía usarse en la oscuridad gracias a la iluminación que tenían implementada. Este sistema de comunicación no era todo lo rápido que se quería, pero conseguía transmitir una señal mediante comunicaciones ópticas durante cientos de kilómetros [11].

El primer experimento de VLC fue gracias a Alexandre Graham Bell que inventó el fonógrafo en 1880. Este instrumento permitía la transmisión del sonido por medio de emisiones de luz. El principio básico del fonógrafo consistía en modular una emisión de luz directamente al receptor que era donde se conectaba el teléfono. La modulación se realizaba por medio de un espejo vibratorio o por un disco rotatorio que periódicamente oscurecían el haz de la luz [12]. Fue el primer instrumento que emitió luz.

Durante la segunda guerra mundial, tanto los del Eje como los Aliados, usaron la tecnología FSO, como por ejemplo, el alemán *Lichtsprechgerät* 80 y el servicio telefónico americano de infrarrojos [11].

En 1955, se introdujo el control remoto de televisión inalámbrico, *Flash-Matic Tuning*. Este sistema sólo estaba compuesto por un botón que permitía encender, apagar o cambiar de canal, todo esto gracias a las células fotoeléctricas situadas en las cuatro esquinas de la pantalla. Esta fototecnología fue una innovación significativa en televisión y permitió la transferencia de señales inalámbricas. Un año más tarde, en 1956, este sistema fue reemplazado por tecnología ultrasonido. Desde entonces, el control remoto IR sigue siendo muy común [12].

Si nos ceñimos a la idea de comunicación mediante la luz visible, nos podemos remontar al siglo XIX, dónde se desarrollaron tanto el heliógrafo como el fonógrafo [13].

El heliógrafo era un artilugio compuesto por un sistema formado por un espejo pivotable o un espejo mediante un obturador que era capaz de pulsar la luz del sol permitiendo la comunicación a largas distancias. Este dispositivo utilizaba código Morse para enviar el mensaje, y alcanzaba distancias de más de 120 kilómetros de distancia; esta tecnología estuvo en uso hasta hace relativamente poco en la armada paquistaní [14]. El mayor inconveniente de este sistema era que precisaba una fuente de luz potente, el Sol o la Luna, y un día despejado; por otro lado, el mensaje podía ser interceptado en cualquier lugar a lo largo de la línea de comunicación, lo cual se trataba de evitar estrechando el haz de luz mediante un tubo en el espejo.

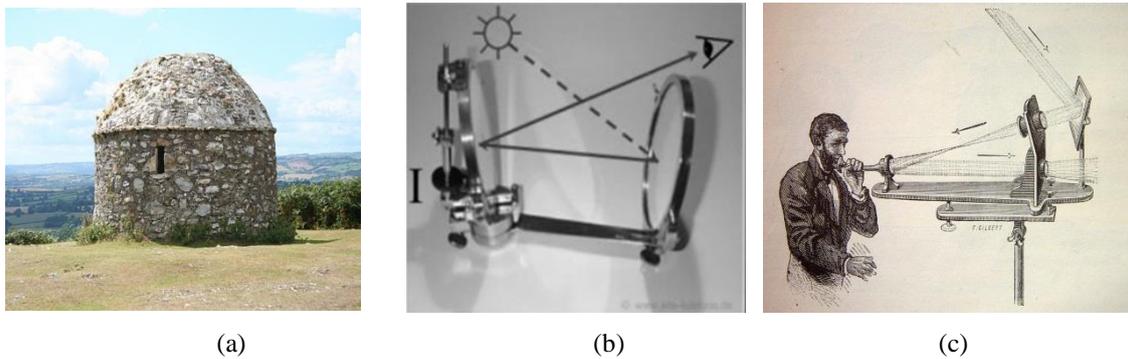
Adicionalmente, el heliotropo utilizaba una tecnología similar para mediciones topográficas, y su uso se extendió hasta la implementación de los modernos sistemas de posicionamiento global GPS.

A finales del siglo XIX, el primer sistema capaz de enviar voz a distancia fue inventado por Alexander Graham Bell junto con su asistente. Conocido como fonógrafo [15], este dispositivo contenía una pieza elástica que reflejaba la luz o la dispersaba en función de las vibraciones producidas por la voz. Así, un haz de luz modulado transmitía la información de la voz y un mecanismo similar en el receptor la demodulaba. Posteriormente, se mejoró la recepción mediante el uso de selenio en el espejo del mecanismo receptor; sus conductividad variable en función de la exposición a la luz convertía la señal óptica en eléctrica, pudiéndose escuchar mediante un micrófono [16]. Mediante este sistema se logró una comunicación clara a una distancia de poco más de 200 metros, pero había problemas ante las interferencias lumínicas naturales.

Su desarrollo posterior por parte, principalmente, del ejército alemán y el británico durante la Primera y Segunda Guerra Mundial, se introdujeron mejoras con nuevos

materiales tanto en el receptor como en el emisor, lo que permitió alcanzar rangos de hasta 15 kilómetros [17].

En la **Figura 2.1** se muestran estos tres sistemas precursores de la comunicación por luz visible actual: una baliza del siglo XVI utilizada para comunicar la llegada de ataques por mar, un operario emitiendo información con un heliógrafo en 1910 en la frontera de Canadá y una ilustración del funcionamiento del transmisor de un fotófono.



**Figura 2.1.** a) Baliza s. XVI (Devon, UK). b) Heliógrafo Mance (1910). c) Modulación en el transmisor de un fotófono.

Una de las primeras tecnologías donde se implantó FSO fue una aplicación militar, utilizada en el ejército francés a finales del siglo XVII. El sistema de télégraphie de Chappe consistía en estructuras de madera de cinco metros de altura separados uno de otros por once kilómetros. Estaba compuesto por tres brazos movibles capaces de articular 196 señales que se traducían por palabras o frases con un significado concreto. A estas estructuras se les incorporaban telescopios para que así, desde la estructura adyacente, se pudiese ver el mensaje transmitido y enviarlo a la siguiente. En un minuto, una señal se desplazaba 135 kilómetros. Este sistema también podía usarse en la oscuridad gracias a la iluminación que tenían implementada. Este sistema de comunicación no era todo lo rápido que se quería, pero conseguía transmitir una señal mediante comunicaciones ópticas durante cientos de kilómetros [12].

El primer experimento de VLC fue gracias a Alexandre Graham Bell que inventó el fotófono en 1880. Este instrumento permitía la transmisión del sonido por medio de emisiones de luz. El principio básico del fotófono consistía en modular una emisión de luz directamente al receptor que era donde se conectaba el teléfono. La modulación se realizaba por medio de un espejo vibratorio o por un disco rotatorio que periódicamente oscurecían el haz de la luz [13]. Fue el primer instrumento que emitió luz.

Durante la segunda guerra mundial, tanto los del Eje como los Aliados, usaron la tecnología FSO, como por ejemplo, el alemán *Lichtsprechgerät* 80 y el servicio telefónico americano de infrarrojos [12].

En 1955, se introdujo el control remoto de televisión inalámbrico, *Flash-Matic Tuning*. Este sistema sólo estaba compuesto por un botón que permitía encender, apagar o cambiar de canal, todo esto gracias a las células fotoeléctricas situadas en las cuatro esquinas de la pantalla. Esta fototecnología fue una innovación significativa en televisión y permitió la transferencia de señales inalámbricas. Un año más tarde, en 1956, este sistema fue reemplazado por tecnología ultrasonido. Desde entonces, el control remoto IR sigue siendo muy común [14].

### 2.2.1 Evolución de las comunicaciones por luz visible

Cuando se habla de VLC, se refiere a una fuente de iluminación, como por ejemplo un LED, que además de alumbrar es capaz de transmitir información utilizando la misma señal de luz. En otras palabras, VLC se puede descomponer como la suma de iluminación más comunicación. Esto es posible gracias a la alta velocidad de transmisión de datos, a la que el ojo humano no es capaz de apreciar destellos de luz [18].

Es una comunicación emergente que nos ofrece una transmisión de datos por medio de sistemas inalámbricos a una alta velocidad, ya que es una característica de los sistemas muy demandada actualmente. Los sistemas VLC, como se ha dicho antes, transmiten datos por medio de luz visible usando el espectro de frecuencia de 400 a 800 THz.

Hay otros términos utilizados en el ámbito de las comunicaciones por luz visible, que tienen un significado parecido a VLC pero con diferencias. Estos tres términos son:

- Comunicaciones **FSO** (*Free Space Optical*). Es parecido a VLC pero no sólo considera la luz visible, sino también la luz ultravioleta y los infrarrojos. Además no se necesita iluminación para FSO, por lo que tiende a ser utilizada en haces de luz estrechos para aplicaciones como enlaces de comunicación entre edificios. FSO a menudo usa láser en lugar de diodos LED [19].
- **LiFi**, este término se explicará más detalladamente a lo largo de este capítulo. Suele usarse para describir aplicaciones de alta velocidad de VLC, en escenarios donde WiFi podría usarse en lugar de LiFi. La tecnología es similar a WiFi con la diferencia que en vez de transmitir por ondas radio, se transmite por medio de la luz.
- **OWC** (*Optical Wireless Communication*). Este término es muy general, ya que se refiere a todo tipo de comunicaciones ópticas inalámbricas. VLC, LiFi, FSO o control remoto por infrarrojos son ejemplos de OWC [19].

Primero se intentó transmitir datos a través de luz incandescente e incluso con lámparas fluorescentes. Estas difícilmente podían transmitir datos, ya que en estas transmisiones había destellos de luz evidente. Además este tipo de bombillas no se pueden encender más de una o dos veces por segundos y no durante mucho tiempo, ya que no pueden soportar ser encendidas y apagadas tantas veces durante mucho tiempo y por eso terminan quemándose. Por lo tanto, FSO a través de estas bombillas habría sido inviable. Además de lo anterior, son ineficaces, ya que se transmitiría a una velocidad muy baja, y sería muy incómodo estar presente en dicha comunicación debido al hecho de una bombilla parpadeante [20].

Gracias al desarrollo del LED, se llega al punto de inflexión en las comunicaciones FSO, ya que nace una nueva subcategoría además de la tecnología de infrarrojos existente. Nació la comunicación por luz visible (VLC) también conocido por luz de red inalámbrica o WiLi.

En la actualidad, el término *Visible Light Communications* (VLC), se refiere a la tecnología de la comunicación que utiliza una fuente de luz visible –habitualmente dispositivos fotónicos de estado sólido como LEDs o láser– como emisor, el aire como medio de transmisión, y un sensor de luz –normalmente un fotodiodo– como receptor.

VLC se refiere a la comunicación de datos en el rango visible del espectro electromagnético. Este rango se mide de 400 a 700 nm de longitud de onda o espectro visible. El término VLC apareció en 2003 cuando un pequeño grupo de personas en la

Universidad de Keio, en Japón, en los laboratorios Nakagawa, comenzó a experimentar con LED y fotodiodos con el fin de lograr una comunicación a través de luz visible [20].

En los laboratorios Nakagawa, junto con algunas de las mayores empresas de tecnología de Japón, como por ejemplo Panasonic, se crea lo que se conoce como el consorcio de VLC. Más tarde, se unió a este consorcio el correspondiente grupo de infrarrojos, la asociación de datos por infrarrojos (IrDA). Desde entonces, se han realizado muchas actividades de investigación relacionadas con VLC desde todo el mundo, con el programa de marco Europeo OMEGA 7 y el trabajo de la Universidad de Oxford, siendo el más destacado [20].

En 2001, en Praga, se desplegó un nuevo sistema de acceso, RONJA (*Reasonable Optical Near Joint Access*) [21], tecnología controlada por el usuario de un enlace de datos ópticos punto a punto de manera inalámbrica. Este enlace tiene un rango de 1,4 kilómetros y una tasa de datos estable y *fullduplex* de 10Mbps. RONJA puede ser de uso cotidiano, ya que se puede montar en una casa conectándolo a un ordenador o a cualquier otro dispositivo con acceso a la red. Una ventaja de esta tecnología es el bajo coste que tiene. Se puede considerar que es uno de los sistemas inalámbricos de menor precio que existe en el mercado [22,23].

En 2003, en Japón, en la universidad de Keio, se empezó a utilizar el LED como fuente de transmisión de datos por luz visible. Desde entonces, ha sido objeto de numerosas investigaciones para así avanzar en las comunicaciones de luz visible.

En 2006 los investigadores de CCTR PennState (*Center for Information and Communications Technology Research*) proponen una combinación de comunicaciones por línea de potencia (PLC) y la luz blanca del LED para así proporcionar acceso de banda ancha para aplicaciones en interiores. Esta aplicación sugiere que VLC podría ser desplegada en un futuro como una solución perfecta [12].

En 2010, un equipo de Siemens y el Instituto Fraunhofer de Telecomunicaciones demostraron la transmisión a 500 Mb/s con un LED de luz blanca a una distancia de cinco metros y en distancias más largas usando cinco LEDs lograron alcanzar una velocidad de 100 Mb/s [12].

El proceso de normalización de las comunicaciones VLC se realizó en el IEEE en el grupo de trabajo de redes de área personal inalámbricas, en el estándar 802.15. En otro capítulo se desarrollará esta norma más extensamente.

Como conclusión de esta pequeña referencia histórica de algunos de los puntos concretos a lo largo de la existencia de las tecnologías FSO y VLC, se podría decir que las comunicaciones ópticas siempre han sido populares, o bien, consideradas de algún modo. Además pueden ser útiles en aplicaciones muy variadas en un futuro próximo. Teniendo en cuenta todas las ventajas que tienen sobre otros sistemas inalámbricos hoy en día más comunes, se espera que VLC se desarrolle más y sustituya a estas otras.

Actualmente los sistemas VLC, tienen un especial atractivo, ya que pueden ser una solución clave para sistemas inalámbricos, además de ser una solución muy económica.

### 2.3. Estado del arte

Los mayores logros en cuanto a velocidad de transmisión y distancia máxima se han conseguido mediante láser, debido a su menor tiempo de encendido-apagado, y a su menor dispersión de la potencia en una comunicación punto a punto. Actualmente, esta tecnología ya está disponible en módulos listos para su uso que alcanzan los 10 Gbit/s

como solución de última milla. Pero este tipo de tecnología no tiene las ventajas propias del VLC y su uso más habitual está enfocado a sustituir la fibra óptica para ahorrar costes en largas distancias.

En el contexto de proveer comunicación inalámbrica de forma integrada con la iluminación, experimentos han llegado a velocidades de hasta 12.5 Gbit/s, con tasas de error (BER) aceptables, mediante el uso de una lente auto-orientable [24]. De esta forma se reducen las pérdidas de potencia y el ruido, a costa de una mayor complejidad en el receptor.

Para velocidades ligeramente inferiores, 3.4 Gbit/s, se han obtenido buenos resultados mediante la utilización de forma paralela de los tres canales que ofrece un LED de tipo RGB (rojo-verde-azul), y 1.5 Gbit/s con un único canal [25]. Adicionalmente, usando LEDs azules con una capa de fósforo se alcanzan velocidades de 1 Gbit/s usando técnicas de modulación digital por división de frecuencias [26].

El ancho de banda máximo que permite el LED puede ser mejorado mediante técnicas de equalización en el transmisor, consiguiendo aumentar la frecuencia máxima de la fuente en un factor de diez [27].

Otra técnica para aumentar el *throughput* de un sistema VLC sin necesidad de mayor ancho de banda en la fuente de iluminación es mediante la transmisión en paralelo o MIMO (multiple-input multiple-output). Varias pruebas experimentales han demostrado esta técnica con éxito, tanto con sensores de imagen como con sensores de luz [28], incrementando la tasa de transmisión total del sistema en función del número total de transmisores [29].

De todas formas, durante los últimos tres años la mayoría de pruebas y publicaciones hablan de velocidades alrededor de los 100 Mbit/s [30-32] y el primer estándar global publicado sobre VLC en 2011 —IEEE 802.15.7 [5]— admite una velocidad máxima de 96 Mbit/s, lo cual nos permite hacernos una idea de las tasas de transmisión actuales de esta tecnología.

Otra metodología similar, en el sentido que utiliza la infraestructura existente para otro fin, no preparada para su uso en las comunicaciones con altas frecuencias, son las comunicaciones mediante cable eléctrico o PLC (*Power Line Communications*). La integración de estas dos técnicas para dotar de conectividad inalámbrica en interiores ha sido objeto de múltiples estudios [33-35], lo que permite ahorrar costes de implementación y hacer más atractiva su comercialización.

Por otra parte, las características propias de utilizar luz, abre un gran campo de posibilidades más allá de crear una red inalámbrica doméstica.

### 2.4. De radiofrecuencia a VLC

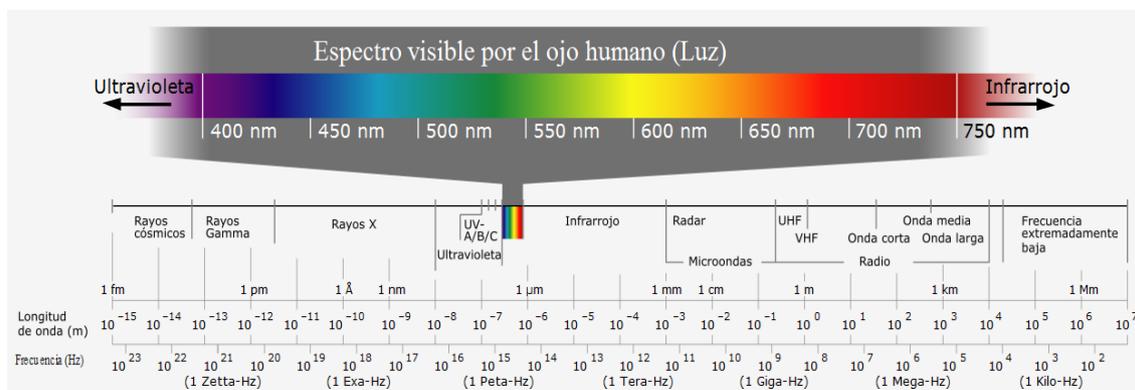
Con el desarrollo del electromagnetismo y las comunicaciones por radiofrecuencias, se mejoró la capacidad de transmitir información a larga distancia, sin necesidad de mantener línea de visión y sin ser percibidas por las personas, relegando a los sistemas de luz visible a un segundo plano en lo que respecta a las telecomunicaciones. Su uso quedó en el ámbito de la señalización, como semáforos, o en paneles de control, dónde el receptor es directamente el ojo humano.

En la última década el escenario ha cambiado progresivamente. En primer lugar, la iluminación de estado sólido ha ido avanzando de forma rápida en su desarrollo y su uso. En 2014 los diodos emisores de luz (LEDs) ya pueden sustituir cualquier fuente de

iluminación artificial tradicional como lámparas fluorescentes, incandescentes o sodio de alta presión (HPS) para todo tipo de aplicaciones, debido a su mayor eficiencia energética (llegando en 2014 a 200 lm/W [36]), vida útil (más de 50 mil horas) y flexibilidad de diseño.

Esto permite la posibilidad de utilizar la infraestructura de iluminación futura con LEDs para las comunicaciones mediante un mínimo de componentes añadidos, dada su capacidad de conmutación a muy alta velocidad (GHz).

Por otra parte, el uso generalizado de las radiofrecuencias ha hecho que la utilización de esta parte del espectro electromagnético tenga que ser regulada; la comunicación visible no está sujeta a estas regulaciones ya que emite por encima de los 3 THz, una banda del espectro libre y gratuita, con capacidad para enviar información en varios canales paralelos sobre todo en espacios interiores. La **Figura 2.2** muestra el espectro de luz visible respecto a la fracción del espectro electromagnético utilizado en radiofrecuencia.



**Figura 2.2.** Espectro radioeléctrico

El hecho de que, a diferencia de los sistemas más comunes basados en RF, no se genere radiación electromagnética, lo convierte en una alternativa atractiva en ámbitos delicados donde la compatibilidad electromagnética es crucial, como hospitales o aeronaves, donde ya se ha planteado su aplicación para poder ofrecer conexión a internet a los pasajeros [37]. En las aeronaves, además, supondría una reducción de costes y peso en cables y apantallamientos; y la misma capacidad de modificar los parámetros de la iluminación LED, necesaria para la comunicación, se podría utilizar para controlar los ciclos circadianos de la tripulación y los pasajeros, es decir mantener a la tripulación alerta y a los pasajeros relajados, aumentando tanto la comodidad de los pasajeros y como la seguridad de la tripulación en vuelos transoceánicos de larga distancia [38].

## 2.5. Características de VLC

Actualmente las tecnologías de comunicaciones inalámbricas son mediante ondas electromagnéticas, la más popular en el uso cotidiano es WiFi. Aunque hoy en día está aumentando la tecnología con LEDs, gracias a su desarrollo, sobre todo para sistemas interiores. Después del conocer en qué consiste esta tecnología y sus antecedentes, se van a explicar ahora las características principales y así después concluir con las ventajas y desventajas de la misma y la comparación con otras tecnologías similares.

- **Visibilidad:** Como el propio nombre de esta tecnología indica, VLC transmite flujo de datos a través de luz visible. Por lo tanto, al tiempo que se emiten datos, existe una iluminación agradable en la habitación in situ.
- **Espectro no regulado:** el rango del espectro electromagnético abarca el rango de 3 Hz a 300 GHz. Existen dos tipos distintos de licencias de espectro de comunicación, la concesión de licencias de radiodifusión y la concesión de licencia celular.

En todo el espectro hay un rango especial que no necesita licencia, ICM (Industrial, Científica y Médica). VLC pertenece a este rango y va desde 370 THz a 870 THz. Por lo tanto, no hay problema por el uso del espectro ya que no necesita licencia [39].

- **Seguridad humana:** Con respecto la transmisión a través de luz, se puede encontrar problemas de seguridad, sobre todo para el ojo. La modulación PHY de VLC puede hacer que la luz parpadee. El parpadeo de la luz puede tener efectos nocivos para la salud en los animales o personas.

El destello de luz da impresión de inestabilidad visual inducida por un estímulo de la luz cuya distribución espectral fluctuará con el tiempo. El CFF (*Critical Fusion Frequency*, o umbral de fusión de parpadeo) es la frecuencia en la que un estímulo de luz intermitente parece ser completamente estable para el observador. Un parpadeo en la luz causa impactos nocivos en la salud del observador como, por ejemplo, molestias visuales, fatiga visual, dolor de cabeza, disminución en el rendimiento de las tareas mentales como comprensión lectora, etc. [40].

Los sistemas VLC, a pesar de la tasa de transmisión de datos, tienen que cumplir el requisito de que la frecuencia de modulación debe ser mayor que el umbral CFF para así evitar daños perjudiciales para la salud. Las siguientes normas regulan el parpadeo de la luz para estos sistemas [41].

- **IEC 1000-3-3/ EN 61000-3-3:** Compatibilidad electromagnética. Parte 3-Límites- Sección 3: Limitación de fluctuaciones de voltajes y parpadeo en redes de suministro de energía de bajo voltaje para equipos con corriente de entrada menor de 16 A.
- **IEC 1000-3-5:** Compatibilidad electromagnética. Parte 3- Límites. Sección 5: Limitación de fluctuaciones de voltajes y parpadeo en redes de suministro de energía de bajo voltaje para equipos con corriente de entrada mayor de 16 A.

La potencia del rango del espectro visible más cercano a la radiación láser infrarrojo 8400-1400 nm penetrará en el globo ocular y puede producir desprendimiento de la retina. Las normas relacionadas con esto se encuentran en [42]:

- **IEC 60825:** Seguridad del láser.
- **ANSI Z136:** Normas de seguridad del láser.
- **EN 207:** Norma europea para la protección de seguridad con el láser.

Los LEDs usados como lámparas tienen que ser evaluados y clasificados para que cumplan las normas de seguridad anteriores. Por eso se dice que las

comunicaciones por luz visible no son nocivas para la salud, ya que se aplican los requisitos de seguridad mencionados y así no dañan los ojos.

- **Uso en área restringida:** En muchas ocasiones, los dispositivos de comunicación de radiofrecuencia se tienen que apagar ya que están en lugares donde pueden causar interferencias, como por ejemplo en los vuelos o en los hospitales. Esto, con el uso de VLC, no hace falta ya que no causa interferencias. Además se puede usar en lugares donde hay población más sensible a la salud como hospitales, colegios o residencias de ancianos ya que esta tecnología no tiene efectos adversos en la salud [43].
- **Alta tasa de datos:** VLC es un tipo de comunicación óptica inalámbrica. Esta tecnología tiene altas velocidades, por lo tanto, las comunicaciones por luz visible heredan esta característica [44].
- **Ancho de banda:** Como se ha dicho antes, las comunicaciones por luz visible explotan la región visible del espectro electromagnético que abarca de 3 kHz a 300 GHz. Sin embargo las comunicaciones de luz visible abarcan el rango de 400 a 800 THz. Por lo tanto la banda de frecuencia utilizada en los sistemas VLC es 1000 veces mayor que la usada en las comunicaciones de radio frecuencia.
- **Naturaleza ubicua:** Cuando se habla de ubicuo se refiere a que está presente en todas partes al mismo tiempo. En estos sistemas se tiene una infraestructura de luz bien establecida hacia el mundo exterior. Además de esto, los dispositivos de iluminación a base de bombillas LEDs son cada vez más aceptados y por lo tanto más incorporados en los sistemas luminarios a nivel mundial. Desde la existencia de la tecnología VLC el uso de fuentes de luz visible para las comunicaciones inalámbricas es más frecuente, así que se espera que se convierta en una tecnología ubicua en un futuro cercano [45].
- **Seguridad:** Debido a que los datos se transmiten por medio de la luz, respecto a la seguridad de los datos en estos sistemas, al no haber ondas electromagnéticas de por medio, tanto la interceptación de la señal como la interrupción de la misma es casi imposible. Por lo tanto nadie que esté fuera de la habitación donde se está haciendo el intercambio de información puede acceder a los datos que se están transmitiendo. Además de esto, VLC implica comunicación con visibilidad directa, por lo que es posible aprovechar la comunicación sin romper el enlace. Por lo tanto, ofrece una comunicación muy segura y se puede utilizar en aplicaciones militares de alta seguridad donde la comunicación por radiofrecuencia es propensa a intrusos. Esto es consecuencia directa de que la luz no puede traspasar paredes o techos, al contrario que las ondas electromagnéticas.
- **Económico:** Para que un sistema VLC exista sólo hace falta una fuente de datos que emita luz y un receptor con un sensor. La instalación de la fuente es igual que la instalación de la luz, es barata además. El precio de los LEDs es muy barato comparado con otros dispositivos de iluminación ya que el consumo de potencia es mínimo y cumplen la función de iluminar a la perfección. Por lo tanto, con el coste de la instalación y de los LEDs el sistema es bastante económico.

## 2.6. Ventajas y desventajas de VLC

Con todo lo expuesto hasta este punto, podemos sacar como conclusión las ventajas e inconvenientes de los sistemas VLC. En la **Tabla 2.1** se resumen los principales aspectos a favor y en contra de esta tecnología.

**Tabla 2.1.** Ventajas y desventajas de VLC

VENTAJAS	INCONVENIENTES
No es nocivo para el cuerpo humano	Absorción atmosférica
Transmisión de datos por tramas de luz existentes	Dispersión del haz de luz
Alivio de los problemas asociados con los sistemas radio de comunicación	Ruidos de interferencias de otras fuentes tienen que ser filtrados
Menor consumo energético	No hay comunicación si no hay visibilidad directa
Incremento de la seguridad	Sólo espectro discreto disponible como fuente de luz y sensor
Integración compacta en sensores de pequeñas dimensiones	Interferencia de fuentes de luz de fondo
Electrónica simple como el controlador del LED	Desvanecimiento de la señal
No tiene influencia sobre otros equipos sencillos de ondas de radio	
Gran número de canales sin interferir con otras fuentes	

Esta tecnología tiene muchas ventajas. Una de las principales es que no satura el espectro usado, ya que usa el espectro visible y este es de mayor capacidad. Además, la información llega por el haz de la luz, por lo tanto se puede manipular la anchura de este, o bien crear un haz disperso que proporcione una cobertura amplia, o, por el contrario, un haz muy fino que ilumine pequeñas zonas y transmita información de forma más concreta y direccional, por lo que se puede decir que la privacidad y seguridad de los datos están aseguradas.

Por otro lado, una de las grandes desventajas está relacionada con lo anterior. Como la luz no sobrepasa objetos o muros, se necesita que haya visibilidad directa entre el transmisor y receptor para que haya comunicación. Esto también condiciona que la cobertura de la comunicación se corte en cuanto se interponga cualquier objeto en el haz de luz. También hay que tener en cuenta que el alcance de esta tecnología es muy reducido. Para aumentar la distancia de transmisión, la potencia de la fuente de la luz debe ser incrementada. El sensor de imagen se puede utilizar en combinación con lentes telescópicas para realizar enlaces de larga distancia. Desafortunadamente, esta mejora lleva un aumento apreciable en el coste de implementación. Se espera que en un futuro esto se mejore sin que aumenten los costes, ya que a más de cinco metros aproximadamente la señal apenas llega [45].

Como la luz no sobrepasa objetos no habrá interferencias con aplicaciones interiores en otra habitación, y al no estar en el mismo rango de frecuencias que las ondas electromagnéticas, no habrá interferencias con equipos que se comuniquen por medio de ondas radio.

Otro de los inconvenientes enunciados es que VLC sufre de desvanecimiento y atenuación atmosférica. Esto limita las aplicaciones de alta velocidad de datos a enlaces

de comunicación de corta distancia. Sin embargo, teniendo un enlace de corta distancia, con suficiente potencia de transmisión este ya no es un problema [46].

También se puede usar para transmitir grandes volúmenes de datos entre equipos o a dispositivos multimedia. Por ejemplo, si se quiere enviar un vídeo del móvil a un televisor de forma rápida o copiarlo a un disco duro de red, bastaría con apuntar el teléfono a la televisión o al disco duro durante unos segundos para que la comunicación se produjese y se transfiriesen los datos, como se hace actualmente con el mando a distancia para cambiar de canal.

## 2.7. Comparación de VLC con otros sistemas inalámbricos RF

Para evaluar la utilidad real del uso de la tecnología VLC en las telecomunicaciones, compararemos las características básicas de este sistema con la tecnología más similar utilizada en el ámbito de las redes inalámbricas de uso personal: las comunicaciones basadas en infrarrojos (IrC), y la tecnología más extendida en su uso actual: las comunicaciones basadas en radiofrecuencia (RF).

### 2.7.1. VLC vs IrC

La radiación infrarroja, también llamada radiación térmica, es un tipo de radiación electromagnética con una longitud de onda mayor que la del espectro visible pero menor que el espectro de ondas electromagnéticas, por lo tanto, tiene menor frecuencia que la luz visible pero mayor que las microondas [47]. Su longitud de onda oscila entre 700 nm-900 nm, y está justo detrás del espectro de la luz visible del rojo.

Se clasifica en tres métodos según el modo de transmisión, estos son: punto a punto, casi-difuso y difuso [48]. Para las transmisiones punto a punto debe haber visibilidad directa, ya que se hace de forma direccional. El inconveniente de esta transmisión son sus limitaciones, ya que hay que colocar el transmisor y receptor enfrentados. Se suele usar en redes inalámbricas.

En el método casi-difuso, a diferencia que en el método punto a punto, el tipo de emisión, en vez de ser direccional, es radial. Esto significa que se produce en todas las direcciones. Para conseguirlo, se transmite hacia distintas superficies reflectantes, y estas redirigirán hacia las estaciones receptoras. Dependiendo de cómo sea la superficie reflectante, se puede clasificar en dos tipos de reflexión, pasiva y activa. En la reflexión pasiva, debido a las cualidades reflexivas del material, la superficie reflectante sólo refleja la señal. Mientras que en la reflexión activa, el medio reflectante, al que se le conoce como satélite, además de reflejar la señal, también la amplifica. La reflexión pasiva es más barata y flexible pero necesita que la potencia de transmisión sea mayor, ya que no tiene etapa amplificadora.

Por último, en el modo de emisión difuso, la diferencia con el método casi-difuso es que debe ser capaz de abarcar, mediante múltiples reflexiones, todo el recinto en el cual se encuentran las estaciones. Para conseguirlo, se requiere una potencia superior que en los otros dos tipos de transmisiones, ya que las pérdidas son mayores puesto que el número de rebotes inciden directamente en el recorrido de la señal.

Algunas de las aplicaciones más comunes de esta tecnología son el envío de documentos desde un ordenador a una impresora, coordinar agendas o libretas telefónicas entre ordenadores, enviar imágenes desde una cámara a un ordenador, control remoto entre dispositivos electrónicos o aplicaciones médicas.

Las comunicaciones por infrarrojos tienen muchas similitudes con los sistemas VLC, ya que se transmiten a través de luz, tienen una longitud de onda del mismo orden por lo que la cobertura que cubren no es muy amplia, y son sistemas difíciles de interceptar. En la siguiente tabla se muestra una comparación entre ambas tecnologías.

Las comunicaciones por infrarrojos ofrecen las mismas ventajas. En términos de ancho de banda disponible, está por detrás de las comunicaciones por luz visible en ciertas áreas. Como ambos sistemas transmiten en frecuencias poco utilizadas en las aplicaciones, el ancho de banda no se tiene que restringir, por lo tanto el uso del espectro de frecuencias no está regulado.

La tasa de datos en los sistemas IR no se puede aumentar más allá de un nivel prescrito ya que puede ser peligroso para el ojo humano [49]. Este problema se puede solucionar usando VLC, ya que la luz visible no lo daña porque el LED transmite luz a muy baja potencia. Por lo tanto, VLC es capaz de transmitir a velocidades mucho más altas que las comunicaciones por infrarrojos. Como se muestra en la **Tabla 2.2** la tasa de transmisión de los sistemas VLC alcanza más de 100 Mbps, mientras que por infrarrojos se transmite hasta los 4 Mbps en FIR (*Fast Infrared*) y los 16 Mbps (*Very Fast Infrared*).

Las comunicaciones por infrarrojos están estandarizadas por la IrDA (*Infrared Data Association*). IrDA es una asociación sin ánimo de lucro que proporciona normas para garantizar la calidad e interoperabilidad de los equipos infrarrojos. Mientras que los sistemas de comunicación visibles están estandarizados en el grupo 802.15.7 del IEEE.

La cobertura que cubre la transmisión de VLC puede alcanzar varios metros gracias al servicio de la iluminación. Por el contrario, la distancia máxima para un controlador remoto utilizando infrarrojos no alcanza más de tres metros. Por lo tanto, el rango de transmisión es mayor en las comunicaciones por luz visible que por infrarrojos.

**Tabla 2.2.** Comparación entre sistemas VLC e IrC

Constante	VLC	IrC
<b>Tasa de datos</b>	Depende del LED, por lo general mayor de 100 Mbps 100Mbit/s – 1 Gbit/s >1 Gbit/s [24,25]	4 Mbps (FIR) 16 Mbps (VFIR) 1 Gbit/s Giga-IR (en desarrollo)
<b>Tipo de LED</b>	RGB, PLED	IR
<b>Rango</b>	< 5 m	< 3 m
<b>Longitud de onda portadora</b>	380nm -780nm	850-900 nm (IrDA) 930-950 nm (RC)
<b>Fuente de ruido</b>	Luz solar u otras fuentes de iluminación	Luz ambiental
<b>Seguridad</b>	Buena	Buena
<b>Regulación</b>	No	No
<b>Estándares</b>	JEITA1221 JEITA1222 JEITA1223 IEEE802.15.7	Comunicación: IrDA (SIR, MIR, FIR, VFIR, UFIR) RC: Sony (SIRC), Phillips (RC5, RECS80), NEC protocol, etc Estándares para medición y termografías
<b>Riesgos</b>	Radiación inocua, seguro para el ojo	Daños en los ojos
<b>Servicio</b>	Iluminación y comunicaciones	Comunicaciones
<b>Aplicaciones</b>	Comunicación punto-a-punto, broadcast, medición, localización, realidad aumentada, señalización. Comunicaciones interiores y en Vehículos.	Conexión punto-a-punto, medición, control remoto.

Como se aprecia en la **Tabla 2.2**, las longitudes de ondas de ambas comunicaciones son distintas. Esto hace que las fuentes de ruido sean diferentes. En el caso de las comunicaciones por infrarrojos el ruido proviene de la luz ambiental ya que esta contiene luz infrarroja, por el contrario en el caso de VLC las fuentes de ruido son tanto la luz solar como cualquier otra fuente de iluminación.

En ambos casos la seguridad es buena, pero gracias a que la fuente de información de VLC es luz visible, daña menos el ojo que la luz infrarroja. También, aunque en ambos sistemas la fuente es luz, la luz infrarroja no la aprecia el ojo humano. Por lo tanto, uno de los servicios que ofrece VLC es la iluminación además de la comunicación. Las comunicaciones por infrarrojos, al llevar más tiempo usándose, tienen muchas más aplicaciones que la tecnología VLC por ser mucho más reciente.

### 2.7.2. VLC vs RF

Las comunicaciones por radiofrecuencia, llamadas también radiocomunicación es una forma de telecomunicación que se realiza mediante ondas electromagnéticas, cuya longitud de onda va de 1 mm hasta 10000 km, en otras palabras, su frecuencia oscila entre los 3 Hz hasta los 300 GHz. Las propiedades de esta tecnología dependen de la frecuencia a la que se opere, por ello, hay bandas conocidas como baja frecuencia, media frecuencia, alta frecuencia, etc.

Cuando la onda de radio actúa sobre la antena, se crea una corriente eléctrica debido al movimiento de la carga eléctrica. Esta corriente se transformará en señales portadoras de información [50]. El transmisor produce una señal portadora cuyas características serán modificadas dependiendo del tipo de información que emita, como por ejemplo una señal audio o video. Esto es lo que se conoce como modulación. La señal modulada se propaga hasta que el receptor la recibe y la demodula, y así conseguir la señal que emitió el transmisor.

Tiene aplicaciones muy variadas, ya que desde hace muchos años es la forma más común de telecomunicación. Entre sus aplicaciones cabe destacar la televisión, la radio y la telefonía móvil, entre otras.

Aunque hoy en día las comunicaciones de RF sean la tecnología inalámbrica más popular, también presenta inconvenientes. VLC puede ser utilizado en comunicaciones inalámbricas donde los sistemas de RF presenten un rendimiento ineficiente.

Al contrario que las comunicaciones por infrarrojos, esta tecnología es muy diferente de VLC. La longitud de onda de ambas tecnologías son de distinto orden, y esto hace que el comportamiento de la señal sea muy diferente. En la **Tabla 2.3** se mostrarán algunas de las principales diferencias.

El ancho de banda, la interferencia y la tasa de datos son unas de las ventajas más significativas de VLC frente a RF. El ancho de banda en el caso de la tecnología VLC no se tiene que restringir. Por esa razón la velocidad de transmisión es alta, del orden de los cientos de Mbps o Gbps mientras que en las comunicaciones por radiofrecuencia es menor, puede llegar hasta decenas de Mbps aproximadamente.

En un sistema por radiofrecuencia, las frecuencias están reguladas por el Cuadro Nacional de Atribución de Frecuencias, basado en el Reglamento de Radiocomunicación de la ITU, esto quiere decir que el uso de las bandas de frecuencias requiere licencias para su uso en cualquier parte del mundo. Sin embargo, en los sistemas de comunicación por luz visible esto no es necesario ya que la banda de frecuencias del espectro visible no está regulada.

Algunos de los factores de atenuación presentes en VLC son la lluvia, la nieve, la niebla, los relámpagos y la luz solar, entre otros, por lo que estos sistemas son más eficientes en lugares cerrados donde no estén expuestos a condiciones climáticas que no son favorables en la comunicación [51].

El rango de frecuencias de VLC es mucho más amplio que el de RF, como se muestra en la **Tabla 2.3**, ya que para este último abarca 300 GHz aproximadamente frente a los 500 THz que comprende la tecnología VLC. Como se puede ver es hasta diez mil veces mayor, como conclusión, las frecuencias disponibles son mucho mayores en la tecnología por comunicación visible.

**Tabla 2.3.** Comparación entre sistemas VLC e RF

Constante	VLC	RF
Tasa de datos	Depende del LED, por lo general mayor de 100 Mbps 100Mbit/s – 1 Gbit/s >1 Gbit/s [24,25]	IEEE802.15.4: <100kbit/s IEEE802.15.1: <24Mbit/s IEEE802.11: >1Gbit/s
Longitud de onda portadora	380nm -780nm	
Frecuencia	3 Hz – 300 GHz	370 – 870 THz
Cobertura	Corto ( <i>broadcast</i> ) Largo (punto-a-punto)	Medio, hasta Km
Ancho de banda	Ilimitado, 380-780nm	Regulado, BW limitado
Línea de visión	Modo LOS: Sí Modo NLOS: No	No
Fuente de ruido	Luz solar u otras fuentes de iluminación	Aplicaciones del espectro electromagnético
Seguridad	Sí, portadora y mensaje	No de la portadora, sí del mensaje si está encriptado
Regulación	No	Sí
Estándares	JEITA1221 JEITA1222 JEITA1223 IEEE802.15.7	IEEE802.15.4 (ZigBee, WPAN) IEEE 802.15.1 (Bluetooth), IEEE802.11 (Wi-Fi)
Riesgos	No	Sí
EMC	Sí	No
Consumo de energía	Bajo-medio	Wi-Fi: Medio Bluetooth: Bajo IEEE802.15.4: Muy bajo
Coste	Bajo	Alto
Infraestructura	Iluminación, Punto de acceso	Punto de acceso
Traspaso de paredes	No	Sí
Problemas ante la lluvia/niebla/nieve	Sí	No
Movilidad	Limitada	Sí

Una de las desventajas principales de VLC frente a RF es la cobertura, mientras que en VLC no llega a más de los pocos metros, en RF la cobertura puede llegar a ser de orden de kilómetros, como por ejemplo en aplicaciones relacionadas con la televisión o la telefonía móvil. En VLC es necesaria la visión directa entre transmisor y receptor para

que exista comunicación. Por esa razón, la comunicación por luz visible alcanza poco rango de distancia y no se puede comunicar con otras habitaciones, mientras que las ondas electromagnéticas sí atraviesan obstáculos y no necesitan visibilidad directa para que la transmisión sea posible. Sin embargo, gracias a esta característica, con VLC no se puede interceptar información, ya que, a menos que se esté en el lugar donde se produce ese intercambio de información, es imposible conseguir los datos que se transmiten. En cambio en RF esto no sucede, la información se puede interceptar con mucha facilidad.

Debido a que la luz no traspasa las paredes, esta comunicación se podría utilizar en lugares donde haya personas más sensibles como colegios u hospitales o en lugares donde dispositivos RF no se puedan usar, como los aviones, ya que el ruido a estos sistemas sólo lo pueden causar otras fuentes de iluminación y no causan interferencias a los sistemas RF. Por el contrario, en las aplicaciones de radiofrecuencias, al ser tan abundantes y tener un uso muy amplio del espectro electromagnético, el ruido es causado por cualquier aplicación de dicho espectro que tenga una frecuencia cercana a la que el dispositivo esté operando [51].

## 2.8. Aplicaciones de VLC

Además de las comunicaciones punto-a-punto o *broadcast* mencionadas anteriormente, las comunicaciones por luz visible permiten varias aplicaciones que aprovechan los LEDs instalados para la iluminación y las particularidades de usar luz visible como portadora. En el futuro cercano se espera que sigan surgiendo nuevas aplicaciones, además de las que se mencionan a continuación: localización de personas y mercancías en espacios interiores, puntos de acceso a información o conectividad en cabinas de avión.

### ○ Aplicaciones de localización:

Un problema habitual en los sistemas de localización basados en satélites es la localización en espacios interiores. Es precisamente en esta situación dónde el VLC es una alternativa interesante. Varios diseños permiten aprovechar la matriz de emisores que supone el sistema de iluminación de una infraestructura para conseguir una localización lo suficiente precisa según la aplicación, tanto en el plano horizontal como en tres dimensiones, en entornos con múltiples usuarios y obstáculos.

El cálculo de la posición del receptor se hace a partir de los mensajes enviados por la matriz de LEDs, teniendo en cuenta que su posición es estática y conocida de antemano, y son identificados individualmente mediante una etiqueta modulada en la luz que emiten. Este algoritmo puede utilizar el ángulo de incidencia de cada LED [47], el tiempo de recepción por la diferencia de fase [48], los niveles de intensidad recibidos [49,50], o directamente usando el sensor de imagen de un *smartphone* [51].

Esta tecnología de localización puede ser útil en grandes espacios como terminales de aeropuertos, donde el uso de indicaciones tradicionales puede ser en algún caso confuso para pasajeros no habituados.

### ○ Comunicaciones seguras:

Otra situación en la que la comunicación mediante luz visible presenta ventajas ante los medios habituales de radiofrecuencia es en las comunicaciones seguras de corto alcance. Ya que la luz visible tiene una longitud de onda más corta que

la radiofrecuencia, no se propaga más allá de un espacio cerrado, haciendo imposible que una antena externa intercepte las señales, sin necesidad de encriptar la información o complejos protocolos de seguridad. Esto también permite la posibilidad de “reutilizar” las frecuencias utilizadas en habitaciones adyacentes.

- **Comunicaciones subacuáticas:**

El interés creciente en la exploración oceanográfica ha generado la necesidad de trasladar las comunicaciones a distancia al medio subacuático. Dada la mayor atenuación que tiene en el agua, el uso de radiofrecuencia presenta grandes dificultades y se ha optado por técnicas acústicas. Aunque las ondas de sonido sí se propagan con facilidad debajo del agua, fuertes problemas de *multipath* y poco ancho de banda disponible hacen que la velocidad de transmisión sea muy baja comparada con las que encontramos en otros medios. Ante esto, la tecnología VLC ofrece soluciones para medias distancias [52], permitiendo tasas de velocidad mucho más altas mediante el uso de LEDs en el rango de longitudes de onda de 400-600 nm, donde el agua marina tiene una menor atenuación. Por las mismas razones, y utilizando la misma tecnología, pueden realizarse mediciones precisas debajo del agua [53].

- **Automoción y tráfico:**

Otro escenario en el que se podría utilizar el VLC para introducir innovaciones importantes es el del tráfico y la automoción [54]. La comunicación del conductor con las infraestructuras y con los demás conductores mediante la señalización habitual podría evolucionar drásticamente si se optimizara mediante la comunicación directa del coche con la infraestructura y con los otros coches. Situaciones de peligro se podrían anticipar y evitar, y habría más herramientas para la gestión del tráfico, siendo esto posible sin un gran despliegue de infraestructura nueva ya que podría utilizarse el sistema de señalización e iluminación de las carreteras.

Este sistema podría implementarse también en el transporte aéreo, mejorando la seguridad permitiendo la comunicación directa entre aeronaves a través de las luces de navegación.

- **Realidad aumentada:**

Otras posibles aplicaciones podrían darse en el campo de la realidad aumentada. La información captada por la cámara del teléfono móvil y del sistema de posicionamiento global (GPS) se complementarían con la información que podría emitir la iluminación de una parada de autobús o un portal, por ejemplo. En museos y tiendas la iluminación de cada elemento podría aportar datos e información interactiva a los dispositivos personales de forma sencilla y práctica.

- **Comunicaciones satelitales:**

En cuanto a las comunicaciones con diodos láser, se han llevado más allá de crear un canal punto-a-punto de larga distancia para sustituir la fibra óptica en tramos determinados. Se están comenzando a explorar sus posibilidades en cuanto a las comunicaciones espaciales: a finales de 2013 la agencia espacial norteamericana NASA demostró el primer sistema de comunicación espacial bidireccional por láser, conectando la sonda de exploración lunar LADEE a una

distancia aproximada de 385000 kilómetros [55]. A finales del 2014 se aplicó esta tecnología consiguiendo un enlace auto-orientado desde la Estación Espacial Internacional a la Tierra [56].

- **Iluminación inteligente:**

Estos sistemas cooperan y controlan la iluminación en un espacio para atender las necesidades de los usuarios mientras ahorran energía. Los sistemas de iluminación inteligente forman redes de sensores inalámbricos (WSN, *Wireless Sensor Network*) para monitorizar factores como la intensidad de la luz, el color y control de la iluminación por atenuación [57]. Se han propuesto varios sistemas para combinar VLC y sistemas de iluminación inteligente, como por ejemplo, encender las luces en un área determinada solo cuando sea necesario, luz de intensidad baja para ver películas, luz brillante para leer y hacer videoconferencia.

Recientemente Philips diseñó “Hue”, un accesorio de iluminación inteligente que puede controlar el color y la atenuación utilizando un iPhone [58]. El sistema está en fase de desarrollo y planea utilizar GPS para encender automáticamente las luces cuando el usuario llega a casa. Los ordenadores portátiles, *smartphones* o televisiones inteligentes pueden utilizar en gran medida esta tecnología.

- **Power Line Communication (PLC):**

PLC, también conocido como banda ancha de potencia (BoPL, *Broadband over Power Line*) utiliza la línea de potencia para transmitir información, donde las tomas de corriente eléctrica se pueden usar para encender los dispositivos, así como para recibir datos. Aunque esta tecnología existe desde hace tiempo, no ha adquirido importancia a nivel mundial, sino en países de Europa y de Asia [59].

PLC tiene que competir con el precio y la calidad de las líneas de suscripción digital, DSL, pero tiene la ventaja de las líneas de potencia, es decir, pueden alcanzar áreas remotas que DSL no cubre. PLC se ajusta muy bien a VLC, ya que los datos de la línea de potencia se pueden utilizar para modular LEDs directamente y además se puede alcanzar una velocidad de 600 Mbits/s [60].

- **5G:**

Se considera que VLC sea uno de los candidatos fundamentales de las tecnologías de acceso a redes inalámbricas 5G para entornos interiores. La habilidad de explotar la luz visible en la banda de frecuencias no reguladas entre los 375 y 780 nm para la transmisión de datos proporciona algunas ventajas. El haz de luz es más seguro comparado con las ondas radio, no se propaga a través de las paredes y no causa interferencia electromagnética [61].

En las **Figura 2.3** y **2.4** se muestran distintos escenarios y aplicaciones para uso de la tecnología VLC.



Figura 2.3. VLC para aplicaciones de localización en interiores.



Figura 2.4. Diferentes aplicaciones de VLC: comunicaciones a bordo de avión, en oficina/vivienda, en hospitales, en automoción y tráfico, entretenimiento y realidad aumentada.

## 2.9. Retos de VLC

Como hemos podido ver en las aplicaciones, las características propias de los sistemas VLC como son una longitud de onda más corta en la portadora y la dualidad comunicación-iluminación, traen ventajas en muchos escenarios donde puede suponer el

despliegue de la mitad de infraestructura o poder aprovechar la ya existente, o no tener que preocuparse de la compatibilidad electromagnética; pero estas particularidades, y el hecho de ser una tecnología poco madura, también trae consigo una serie de retos y problemas por resolver.

El primero de ellos es proveer de *uplink* o enlace ascendente a un medio que es por su naturaleza *broadcast*. Los requerimientos de iluminación en oficinas y otros recintos aseguran que hay suficiente potencia para la comunicación; la incorporación de un modulador en el sistema de iluminación es suficiente para emitir en todo el rango de cobertura de la luz, y cualquier receptor dotado de un fotosensor podría recibir la información y demodularla. Para que esta comunicación sea bidireccional es necesario crear un segundo canal, punto móvil-infraestructura o punto fijo-infraestructura.

Existen en la actualidad varias estrategias para resolver este reto, solventando los problemas de interferencia entre *uplink* y *downlink*. La primera es utilizar un enlace con RF [62,63]. Esta aproximación cuenta con la ventaja de que es una tecnología conocida y madura, y evita el problema de las interferencias, pero no podría utilizarse en ambientes sensibles al EMC. Otra solución de este tipo es incorporar la tecnología infrarroja en el canal ascendente [64].

Considerando un canal de luz visible en el *uplink*, y de esta manera manteniendo las ventajas intrínsecas del VLC, dos soluciones han sido estudiadas: mediante la técnica *time-division-duplex*, usando *slots* de tiempo entre transmisión y recepción, se reduce significativamente la interferencia por la reflexión [65], o usando diferentes longitudes de onda en el *uplink* y *downlink* y filtros ópticos para aislar las señales [66].

Otra solución potencialmente atractiva es re-modular la luz recibida en el receptor mediante un retroreflector [67], aunque la velocidad que permiten los retroreflectores moduladores disponibles actualmente es baja y su coste es alto.

El otro reto implícito en el uso de la luz visible como forma de comunicación es mantener el enlace cuando no se precisa iluminación. Como en el canal ascendente, una primera aproximación es la tecnología híbrida, con RF o IR, aunque esto aumentaría la complejidad y el coste del sistema y no sería válido en todos los escenarios. Mediante el uso de un sensor con amplio rango espectral, se podría minimizar los riesgos del uso de IR emitiendo la potencia mínima necesaria para complementar la luz visible y mantener la robustez del canal; los fotodiodos de avalancha son la mejor opción para detectar niveles muy bajos de luz por su ganancia interna.

Si queremos mantener un canal estrictamente basado en luz visible debemos encarar el problema desde el uso de técnicas de modulación eficientes; considerando unos niveles mínimos de potencia admitidos como *apagado* en función del escenario. En posteriores apartados se abordará el tema de la modulación de las fuentes de luz, considerando el *dimming* o regulación de la intensidad de la fuente de iluminación. En general, si está en uso todo el ancho de banda disponible con la luminaria regulada al 100%, bajar los niveles de iluminación supondría bajar la tasa de transmisión proporcionalmente; disponiendo de más ancho de banda, se puede mantener la robustez y la velocidad de transmisión con el control de *dimming*.

Se ha demostrado que utilizando técnicas de modulación digital de pulsos es factible alcanzar tasas máximas de transmisión del orden de 625kbit/s de día, usando un 1% de la potencia máxima de la luminaria, que supone un 0.36% de la luz total; y una tasa máxima de 320kbs usando 0.001% de la potencia máxima de la luminaria, que supone

un 7.5% de la luz que hay durante la noche [68], suficiente para aplicaciones de información y señalización con sensores y actuadores.

El consorcio de las comunicaciones por luz visible (VLCC) se estableció en Japón en noviembre de 2003, con las principales compañías de telecomunicaciones (*Fuji Television*), de iluminación (*NEC Lighting*), de potencia eléctrica (*The Tokyo Electric Power Co.*) de esta ciudad. El objetivo de VLCC es dar a conocer y estandarizar la tecnología de comunicación por luz visible, que ha sido discutida y evaluada en los diversos campos de la industria [3].

En mayo de 2014, tras 11 años de actividad del VLCC, se creó la asociación de comunicaciones por luz visible (VLCA, *Visible Light Communication Association*). VLCA se centra en crear la base del negocio de las comunicaciones por luz visible, ya que actualmente, las luces LEDs se utilizan en una gran variedad de aplicaciones. Por eso quieren establecer una nueva asociación general constituida para apoyar estas actividades. Por este motivo piden a las empresas de varias industrias como la industria de las telecomunicaciones, industria de la iluminación, industria de la infraestructura social, la industria de Internet, la industria informática, la industria de los semiconductores, entre otras, que se unan a la Asociación de Comunicaciones por luz visible para la realización de la infraestructura de comunicación luz visible y su negocio. Se pretende hacer realidad una vida cotidiana cómoda y un ambiente de trabajo mediante el uso de tecnologías de la comunicación por luz visible [69]. Un enlace básico es el mostrado en la Figura 2.5.

### 2.10. Componentes de un sistema VLC

Básicamente en una transmisión VLC nos encontramos con dos componentes fundamentales [2]:

1. **Unidad transmisora.** Genera luz utilizando diversos tipos de LED para emitir luz visible modulada por los datos que se desean transmitir de forma unidireccional/simplex, bidireccional HDX o FDX. Algunos métodos-esquemas de modulación son:
  - (i) **VPPM (Variable Pulse Position Modulation).** Similar al PPM pero permite controlar la anchura del impulso para soportar la bajada de la luz.
  - (ii) **SM (Spatial Modulation).** Permiten determinar la fuente de una señal óptica. Si se puede determinar su fuente puede utilizar múltiples fuentes de información para transportar múltiples flujos de datos independientes (uno de cada fuente) o se puede emplear la fuente de la señal como parte de la propia codificación de la información. Varias fuentes pueden ser varios LEDs dentro de un único dispositivo o luminaria.
  - (iii) **OFDM (Orthogonal Frequency Division Multiplex).** Utiliza un conjunto de sub-portadoras cada una en diferentes frecuencias pero armónicamente relacionadas. Presenta una buena eficiencia espectral pero es algo complejo de implementar.
  - (iv) **PWM (Pulse Width Modulation).** Permite transportar la información codificada en la duración de los impulsos. Se pueden transportar más de un bit de datos en cada impulso, pero pueden ser necesarios mayores las longitudes de los impulsos que para OOK, de modo que no presenta grandes ventajas. Es fácil de implementar.

(v) **CSK (Colour Shift Keying)**. Se utiliza en sistemas que emplean LED-RGB. Combinando los diferentes colores de la luz, los datos de salida se pueden transportar por medio del color en sí, de modo que la intensidad de la salida puede ser constante. Su implementación es compleja.

(vi) **PPM (Pulse Position Modulation)**. Los datos se codifican utilizando la posición del impulso dentro de una trama. Se pueden enviar más de un bit en cada impulso, sin embargo la duración de la trama debe ser mayor que para un único bit OOK, de modo que no necesariamente es más eficiente. En cada trama se contiene la misma cantidad de energía óptica.

(vii) **OOK (On-Off Keying)**. Los datos se generan encendiendo y apagando el LED. El uno lógico se consigue poniendo el LED a ON y un cero lógico se obtiene poniendo en OFF el LED. Es simple de implantar pero no es óptimo en lo relativo a control de iluminación y caudal de datos.

(viii) **PAM (Pulse Amplitude Modulation)**. La información se transporta en la amplitud del impulso. PAM puede transportar más datos en cada impulso que OOK pero es más susceptible al ruido del canal óptico. Por ejemplo en la modulación 256-ASK cada elemento de señal transporta logaritmo en base dos de 256 cuyo resultado es 8 bits. (ix) Otros esquemas son 512-QAM (mezcla ASK en amplitud y PSK en fase), 1024-PSK, BFSK (Binary Frequency Shift Keying) en frecuencia, SS-kPPM (SubCarrier-Pulse Position Modulation) donde k es un número potencia de dos, por ejemplo dos o cuatro. Los datos se representan por presencia y ausencia de la onda de la portadora o carrier.

2. **Unidad receptora.** Utilizan diversos elementos para demodular/decodificar la luz en datos recibidos, se utilizan foto-detectores como fotodiodos, matrices CCD, fototransistores, etc. y mecanismos criptográficos-esteganográficos adecuados para desocultar los cifrados visibles y desocultar lo subliminar de los datos no observables/identificables.

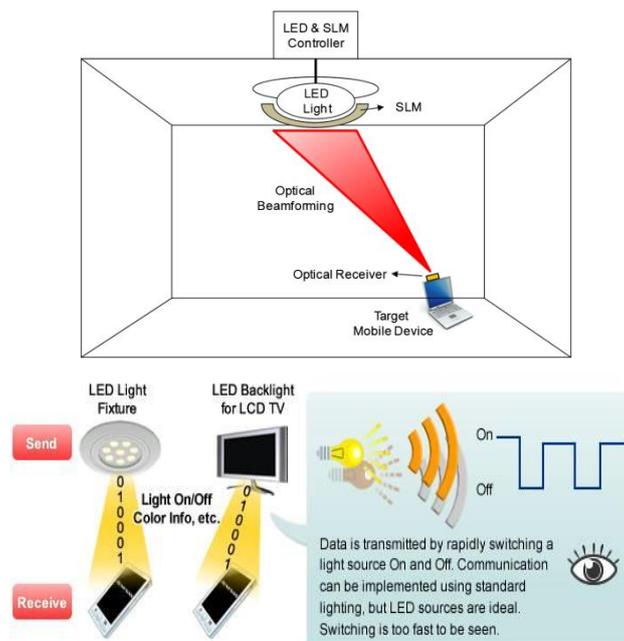


Figura 2.5. Esquema de un sistema VLC básico.

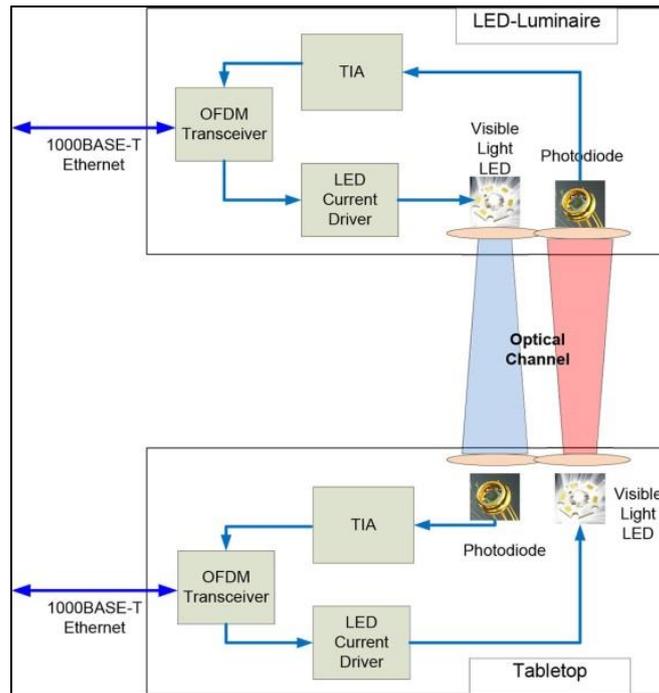


Figura 2.6. Esquema ejemplo de un sistema VLC básico con tecnología OFDM.

## 2.11. Conclusiones

En este capítulo se han descrito los sistemas de comunicación inalámbricos por luz visible, su origen y sus características destacando su doble funcionalidad de iluminación y comunicación de datos de altas velocidades sin conllevar riesgos sobre las personas. De todas las tecnologías expuestas nos hemos centrado en VLC por sus ventajas, siendo objeto de este proyecto. Entre las desventajas que presenta aún por superar destacan el corto alcance y la necesidad de visibilidad directa.

Se ha comparado la tecnología VLC con otros sistemas de comunicación utilizados en la actualidad. Si bien VLC no reemplazará a las tecnologías inalámbricas actuales como WiFi, se podrían complementar, para solucionar las desventajas de ambas tecnologías.

Los sistemas VLC constituyen una solución alternativa que va ligada a la protección del medio ambiente y sobre todo al del ser humano. También destacar que hay multitud de posibles aplicaciones en cualquier situación del día a día.

# Capítulo 3

## El estándar IEEE802.15.7: nivel MAC

---

*Es más fácil hacer leyes que gobernar.*

LEON TOLSTOI (1828-1910)

### 3.1. Introducción

En este capítulo se describen las principales características del estándar IEEE 802.15.7 [5] para redes inalámbricas de área personal, WPAN, denominadas *Visible-light Communication Personal Area Network* (VPAN).

El estudio del estándar IEEE.802.15.7 para VLC, es el paso previo para la implementación de una configuración de red del protocolo. En este capítulo nos centramos en el nivel MAC de dicho estándar. La subcapa MAC realiza funciones de control y gestión de tramas de los distintos protocolos, incluyendo salidas hacia la subcapa física PHY.

### 3.2. Organismos de normalización y estándares

Los estándares son el resultado de un esfuerzo por evitar la fragmentación resultante del desarrollo de soluciones propietarias, que la experiencia demuestra que ocurre cuando una tecnología no se regulariza o estandariza. En este sentido, las comunicaciones ópticas con luz visible es una tecnología con pocas normas establecidas, y son varios los agentes que están fomentando este tipo de comunicaciones. Entre otros podemos nombrar el *Visible Light Communication Committee* (VLCC), *IEEE*, *Light Fidelity* (Li-Fi) o *Wireless World Research Forum* (WWRF), así como grupos investigadores privados y marcas comerciales.

Japón fue pionera en introducir la tecnología VLC a través del VLCC [3], fundado en 2003 y formado mayoritariamente por empresas tecnológicas japonesas. Éste promueve el uso de la luz visible para la transmisión de datos, realizando tareas de investigación, desarrollo, difusión y estandarización.

El VLCC jugó un papel clave en la especificación de dos estándares del Japan Electronics and Information Technology Industries Association (JEITA) en 2003 [70]:

- **CP-1221:** Pensado originariamente para ser usado en la transmisión de información de identificación a bajas velocidades, del orden de 4,8Kbps usando modulación de pulsos.
- **CP-1222:** Destinado a redes de comunicaciones de datos y con especificaciones más restringidas: sugiere esquema de modulación 4-PPM con una frecuencia de subportadora de pulsos de 28.8Kbps. Admite velocidades desde unos pocos Kbps hasta 1Mbps.

En octubre de 2008 el VLCC comenzó a trabajar en cooperación con la IrDA (*Infrared Data Association*) y la ICSA (*Infrared Communication Systems Association*), anunciándose en marzo de 2009 un estándar que adoptaba y expandía la capa física del estándar IrDA. La ICSA ha continuado los trabajos para desarrollar un estándar de red

LAN (*Local Area Network*) que permita transmisión *full duplex*, mediante el uso de multiplexación WDM (*Wavelength Division Multiplexing*) usando las bandas de luz visible e infrarroja.

En el año 2007 se formó el grupo de trabajo para VLC (WG4021) de la TTA (*Telecommunications Technology Association of Korea*) [70], que desarrolló la especificación del estándar TTA 5 en 2008. Esta especificación abarca aspectos como la configuración de la capa física del transmisor y receptor, la configuración básica para la interfaz LED para iluminación y VLC, y la configuración del modelo de servicio de localización basada en luz. Este grupo de trabajo ha desarrollado otros estándares, como el TTA 16, cuyas especificaciones se centran en la convergencia de las tecnologías VLC y los sistemas de iluminación basados en LED.

En 2008 se fundó el proyecto OMEGA (*hOME Gigabit Access*), que cuenta con la cofinanciación de la Unión Europea y la participación de diversas entidades europeas del ámbito académico y de la industria. El objetivo de este proyecto es el desarrollo de estándares para redes domésticas que proporcionen servicios y suministren contenidos a velocidades de transmisión de hasta 1 Gbps, incluyéndose el uso de tecnologías de comunicación óptica inalámbrica en las bandas de infrarrojo y luz visible [71].

A pesar de existir iniciativas de normalización como las citadas anteriormente en el mundo occidental es el IEEE [72] el principal referente de estandarización en tecnologías. Dentro del grupo de trabajo IEEE 802.15, que está dedicado a la normalización de las redes inalámbricas de área personal WPAN (*Wireless Personal Area Network*), en el año 2008 se creó el grupo de interés en VLC conocido como IEEE 802.15.7.

A principios de 2009, el grupo IEEE 802.15.7 comenzó a trabajar en la propuesta de un estándar de VLC que incluyera especificaciones para la capa física y la capa MAC. En el año 2011 se publicó el estándar IEEE 802.15.7-2011 [1] que define el uso de VLC para redes WPAN, en el que se especifica la capa física y la capa MAC para comunicaciones ópticas inalámbricas de corto alcance usando luz visible, con velocidades de transmisión desde 11 Kbps hasta 96 Mbps.

En el año 2011 se funda en Oslo el Li-Fi (*Light-Fidelity*) Consortium [73] con el objetivo de promover el desarrollo y la distribución de tecnologías ópticas inalámbricas en las comunicaciones, la navegación, y otros campos de aplicación. En los últimos años se ha fortalecido el concepto de *Li-Fi* que ha surgido en los últimos años. Bajo esta denominación varios autores, entre ellos el profesor Harald Haas [42] de la Universidad de Edinburgo, quienes han pretendido impulsar la integración del conjunto de tecnologías y servicios relacionados con las comunicaciones ópticas inalámbricas basadas en luz visible dentro de un único concepto. Por tanto, *Li-Fi* no es sinónimo de ninguna tecnología particular sino que se puede considerar un superconjunto de muchas, entre ellas VLC.

En el año 2013 la JEITA publicó el estándar CP-1223, propuesto como una versión simplificada y mejorada del estándar CP-1222. En mayo de 2014 se constituyó la VLCA (*Visible Light Communications Association*) [69] como organismo sucesor del VLCC, para continuar con la investigación, el desarrollo y la estandarización de las tecnologías VLC, contando con el apoyo de empresas líderes del sector de Japón.

En el año 2015, el IEEE anunció la formación de un grupo de trabajo, el IEEE P802.15.7r1 [74] creado para elaborar una revisión del estándar IEEE 802.15.7-2011 que permita el uso de longitudes de onda del infrarrojo y el ultravioleta cercano además

de la luz visible. Asimismo, se ha propuesto la inclusión de varias opciones adicionales que se resumen a continuación:

- Dotar a las cámaras fotográficas con capacidad de comunicación para aplicaciones de localización y posicionamiento, empleando distintas velocidades de transmisión. Para ello, se propone el uso del flash de la cámara como dispositivo transmisor y el sensor de imagen como dispositivo receptor.
- Soporte para sistemas inalámbricos de identificación basados en iluminación LED (LED-ID).
- Capacidad para comunicaciones inalámbricas *Li-Fi* bidireccionales con velocidades de, al menos, 10 Mbps, y soportando comunicación en red con movilidad e itinerancia (*roaming*) entre múltiples puntos de acceso.

Se prevé que el grupo de trabajo IEEE P802.15.7r1 elabore un primer borrador de la nueva revisión del estándar a finales del año 2017 [74].

### 3.3 El estándar IEEE 802.15.7

El objetivo de la norma IEEE 802.15.7 es proporcionar un estándar para comunicaciones ópticas inalámbricas de corto alcance basadas en luz visible, haciendo uso del espacio libre como medio de propagación, para lo que define una capa física (PHY) y una capa de control de acceso al medio (MAC). Las especificaciones de la capa PHY y MAC son capaces de proporcionar velocidades de datos desde los 11Kbps hasta 96Mbps. Usa esquemas de modulación OOK, VPPM y CSK. El estándar IEEE802.15.7 provee velocidades de transmisión suficientes para soportar servicios de comunicaciones multimedia, proporciona soporte para la movilidad, y ofrece compatibilidad con las infraestructuras de iluminación, cumpliendo también con las normativas aplicables en cuanto a seguridad ocular.

Algunas de las características más importantes que se pueden destacar del estándar son:

1. Operación en tres topologías de red: estrella, *peer-to-peer* y difusión.
2. Transmisiones programadas o realizadas mediante técnicas de acceso al medio con contención.
3. Transferencias de datos fiables mediante el uso de tramas de confirmación.
4. Indicación de la calidad del canal óptico, en inglés *Wavelength Quality Indication* (WQI).
5. Soporte para el control de la luminosidad (*dimming*).
6. Soporte para la función de iluminación.
7. Soporte para funciones basadas en el uso del color.

A continuación analizamos cinco cuestiones básicas para profundizar en la comprensión del estándar: tipos de dispositivos, topologías de red, arquitectura de un dispositivo IEEE 802.15.7, el concepto de primitivas y seguridad en VPAN.

#### 3.3.1 Tipos de dispositivos

El estándar IEEE802.15.7 clasifica los dispositivos en tres tipos: infraestructura (fijos), móviles, y vehículos. Las características de cada tipo de dispositivo se indican en la **Tabla 3.1**. En el entorno de las redes de datos en interiores, en el que se definen esta norma, podemos reducir la clasificación a dispositivos infraestructura (normalmente iluminarias) y dispositivos móviles (normalmente dispositivos portátiles de usuario).

Tabla 3.1. Tipos de dispositivos y sus características (IEEE 802.15.7)

Característica	Infraestructura	Móvil	Vehículo
Coordinador fijo	Sí	No	No
Fuente de alimentación	Ilimitada	Limitada	Moderada
Factor de forma	No restringido	Restringido	No restringido
Fuente de luz	Intensa	Débil	Intensa
Movilidad física	No	Sí	Sí
Alcance	Corto/largo	Corto	Largo
Velocidad de transmisión	Alta/baja	Baja	Alta

### 3.3.2 Topologías de red

El estándar IEEE802.15.7 para redes de comunicaciones visibles de área personal (VPAN o *Visible Personal Area Network*) admite tres topologías de red, en las cuales se podrán enmarcar las posibles aplicaciones previstas de sistemas haciendo uso dicho estándar: estrella, *peer-to-peer* y difusión. La estructura básica de dichas topologías se ilustra en la **Figura 3.1**, resumiéndose a continuación sus principales características:

- **Estrella:** la comunicación se realiza entre varios dispositivos y un único controlador central, al que se denomina coordinador. Normalmente, en esta topología el coordinador es un dispositivo de infraestructura de iluminación, conectado a la red eléctrica, y con capacidad para transmitir y recibir datos. El resto de dispositivos son terminales móviles alimentados por baterías. Cada red en estrella tiene un identificador de VPAN único que establece el coordinador, al que deben asociarse los dispositivos móviles para poder comunicarse. Cada VPAN opera de forma independiente a otras VPAN que puedan existir en su mismo ámbito de cobertura.
- **Peer-to-peer:** permite a dos dispositivos cualesquiera comunicarse entre sí, siempre que estén al alcance uno de otro. Uno de los dispositivos debe actuar como coordinador de la comunicación.
- **Difusión:** un dispositivo hace de coordinador y transmite su señal a otros dispositivos de forma unidireccional, sin necesidad de que haya ningún proceso de asociación. Como se ha indicado antes, además de las topologías punto a punto y en estrella, los dispositivos que cumplen con este estándar también pueden operar en la topología de difusión mientras no estén asociados a ningún dispositivo o no tengan ningún dispositivo asociado a él.

Todas las topologías proporcionan soporte de visibilidad, de forma que la función de iluminación se mantenga en ausencia de comunicación o en los modos de funcionamiento en recepción o inactividad

Cada red VPAN independiente posee un identificador único. Igualmente la formación de la red es realizada desde las capas superiores haciendo uso de los servicios de gestión que ofrece la capa MAC, pero tampoco está dentro del alcance de este estándar.

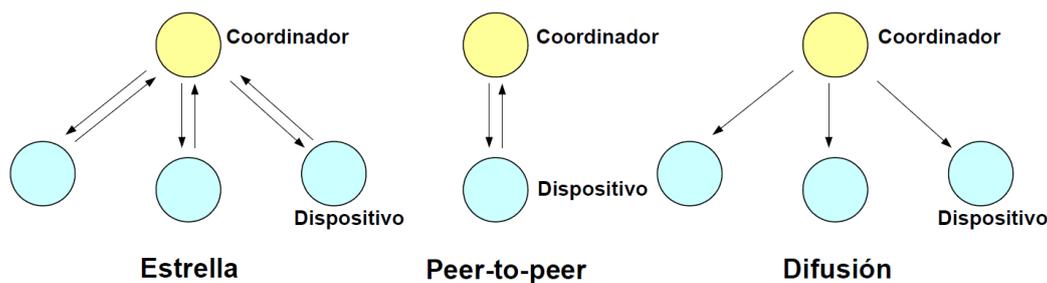


Figura 3.1. Topologías de red en el estándar IEEE 802.15.7

### 3.3.3 Arquitectura de un dispositivo IEEE 802.15.7

La arquitectura definida por esta norma se compone de dos capas cuya especificación es objeto del estándar: una capa física (PHY), que incluye al transceptor óptico, y una capa de control de acceso al medio (MAC), que proporciona acceso al medio o canal óptico.

En la definición de la arquitectura se incluyen también varias capas adicionales. La **Figura 3.2** muestra de forma gráfica el conjunto de capas de la arquitectura. A continuación se describen brevemente las distintas capas definidas en la arquitectura:

- **Capa PHY:** la capa física (PHY) proporciona la interfaz entre la capa MAC y el canal óptico, y es proveedora de dos servicios: el servicio de datos, al que se accede a través del PD-SAP (*PHY Data Service Access Point*), y el servicio de gestión, al que se accede mediante el PLME-SAP (*PLME Service Access Point*). La entidad PLME (*Physical Layer Management Entity*) proporciona el servicio de gestión, siendo responsable de mantener una base de datos con los objetos gestionados por la capa PHY. El PHY-switch hace de interfaz entre la capa PHY y el punto de acceso al servicio *optical SAP* para la conexión con el medio óptico.
- **Medio óptico:** los transmisores y receptores ópticos del dispositivo se consideran parte del medio óptico. Cuando hay más de un transmisor y receptor, la capa PHY permite dar soporte a la comunicación en múltiples celdas. El *optical SAP (Service Access Point)* provee la interfaz de acceso al medio óptico, aunque no es objeto de especificación del estándar.
- **Capa LLC y subcapa SSCS:** la capa LLC (*Logical Link Control*) implementa el modo de operación IEEE 802.2 Tipo 1 [75] y proporciona la interfaz con la capa de red. La subcapa SSCS (*Service-Specific Convergence Sublayer*) [5] proporciona el acceso de la capa LLC a los servicios de la capa MAC.
- **Capa MAC:** proporciona el servicio de datos para la transmisión y recepción de tramas MPDUs (*MAC Protocol Data Units*) a través del servicio de datos de la capa PHY, y el servicio de gestión, que proporciona funciones como la gestión de tramas baliza (*beacon*), el acceso al canal de transmisión, y la asociación/disociación de dispositivos. También proporciona funciones como el soporte de visibilidad y el control de la luminosidad. Para acceder al servicio de datos de la capa MAC se utiliza el MCPS-SAP (*MAC Common Part Sublayer Service Access Point*), y para acceder al servicio de gestión el MLME-SAP (*MAC Sublayer Management Entity Service Access Point*).

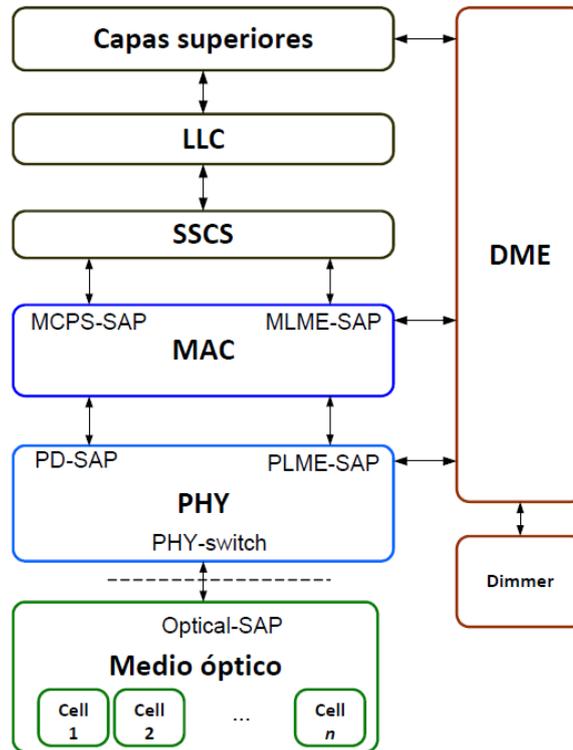


Figura 3.2. Arquitectura de un dispositivo IEEE 802.15.7

- **Capas superiores:** entre éstas se incluye a la capa de red - que incluye funciones como la configuración de la red y el encaminamiento - y la capa de aplicación, que proporciona la función para la que el dispositivo ha sido diseñado. La definición de estas capas no concierne al estándar IEEE 802.15.7.
- **DME y dimmer:** el estándar soporta la existencia de un DME (*Device Management Entity*) que hace de interfaz entre los servicios de gestión de las capas PHY y MAC y un dispositivo regulador de luminosidad (*dimmer*).

En el apartado 3.4 profundizaremos en el estudio de la capa MAC, su estructura y funcionalidad.

### 3.3.4 Modelo de transferencia de datos en IEEE 802.15.7

La arquitectura definida Existen tres tipos de transacciones de transferencia de datos:

- El primer tipo es la transferencia de datos de un dispositivo a un coordinador.
- El segundo tipo es la transferencia de datos de un coordinador a un dispositivo.
- El tercer tipo es la transferencia de datos entre dos dispositivos iguales.

En la topología en estrella o en difusión solo se utilizan los dos primeros tipos, mientras que en la topología punto a punto se pueden utilizar los tres tipos.

El mecanismo utilizado para cada uno de los tipos de transferencia depende de si la red soporta la transmisión de balizas. Se seleccionará el uso de VPANs que permiten baliza en caso de que se requiera sincronización o soporte para dispositivos de baja latencia. En caso contrario, se puede seleccionar la transferencia normal sin el uso de baliza. Sin embargo, la baliza se requiere para la detección de redes.

### 3.3.4.1 Transferencia de datos a un coordinador

Cuando un dispositivo desea transmitir datos a un coordinador en una VPAN que permite el uso de baliza, primero realiza un proceso de escucha del canal para encontrar la baliza. Cuando se encuentra, el dispositivo se sincroniza con la estructura de supertrama. En el instante de tiempo adecuado, el dispositivo transmite su trama de datos al coordinador, utilizando para ello acceso aleatorio ranurado. El coordinador podrá asentir la correcta recepción de los datos transmitiendo una trama de asentimiento.

Cuando un dispositivo desea transferir datos en una VPAN que no permite el uso de baliza, este transmite su trama de datos al coordinador utilizando acceso aleatorio no ranurado. El coordinador asiente la correcta recepción de los datos transmitiendo una trama de asentimiento opcional. En el momento en el que se reciba la trama de asentimiento, la transacción habrá concluido.

### 3.3.4.2 Transmisión de datos desde un coordinador

Cuando un coordinador desea transferir datos a un dispositivo en una VPAN que permite el uso de baliza, este indica en la baliza que está pendiente de enviar el mensaje de datos. Los dispositivos escuchan periódicamente las balizas que envía el coordinador y, si está pendiente de recibir el mensaje, transmite una instrucción MAC requiriendo el envío de los datos utilizando acceso aleatorio ranurado. El coordinador asiente la correcta recepción de los datos transmitiendo una trama de asentimiento. La trama de datos pendiente se envía usando acceso aleatorio ranurado o, si es posible, se envía inmediatamente después de recibir el asentimiento. El dispositivo puede asentir la recepción de los datos transmitiendo una trama de asentimiento opcional. Tras la finalización con éxito de la transacción de datos, el mensaje se elimina de la lista de mensajes pendientes en la baliza.

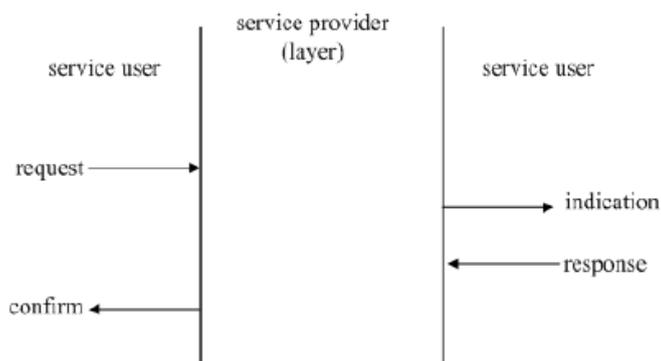
Cuando un coordinador desea transferir datos a un dispositivo en una VPAN que no permite el uso de baliza, almacena los datos y espera establecer el contacto con el dispositivo para que le realice la petición de datos. Los dispositivos pueden establecer el contacto transmitiendo una instrucción MAC en la que se solicite al coordinador el envío de datos, utilizando para ello acceso aleatorio no ranurado. El coordinador asiente la correcta recepción de los datos transmitiendo una trama de asentimiento. Si la trama de datos está pendiente de envío, el coordinador transmite la trama de datos utilizando acceso aleatorio no ranurado. Si la trama de datos no está pendiente de envío el coordinador lo indica ya sea en la trama de asentimiento que sigue a la petición de datos o en la trama de datos con una carga útil o *payload* de tamaño cero. Si se pide, los dispositivos asentirán la recepción de la trama de datos mediante la transmisión de una trama de asentimiento.

### 3.3.4.3 Transferencia de datos punto a punto

En una VPAN con una topología punto a punto, cada dispositivo puede comunicarse con cualquier otro dispositivo que esté dentro de su área de cobertura. Para hacerlo de forma efectiva, los dispositivos que quieran comunicarse deberán permanecer en un estado de recepción constante o sincronizarse con los otros dispositivos. En el primer caso, los dispositivos pueden transmitir sus datos utilizando acceso aleatorio no ranurado. En último caso, se necesitan tomar otras medidas para conseguir la sincronización.

### 3.3.5 Concepto de primitivas

Con el fin de comprender la funcionalidad de la norma IEEE802.15.7 es fundamental abordar el concepto de primitivas de servicio. Los servicios que proporciona una capa son las capacidades que ofrece a la capa superior o subcapa más próxima realizando sus funciones gracias a los servicios de la capa inferior. Este concepto se ilustra en la **Figura 3.3**. Topologías de red en el estándar IEEE 802.15.7, mostrando la jerarquía de servicios y la relación de los dos usuarios correspondientes y sus entidades de protocolos pares de capas (o subcapas) asociadas.



**Figura 3.3.** Concepto de primitiva de servicio.

Los servicios se especifican describiendo el flujo de información entre el usuario y la capa. Este flujo de información se modela mediante eventos instantáneos y discretos, que caracterizan el servicio proporcionado. Cada evento consiste en pasar una primitiva de servicio de una capa a otra a través de un punto de acceso al servicio (SAP) asociado con un usuario. Estas primitivas son una abstracción porque especifican solo el servicio prestado en lugar de los medios por los que se proporcionó.

Los servicios se especifican describiendo las primitivas del mismo y los parámetros que las caracterizan. Un servicio puede tener una o más primitivas relacionadas que constituyen la actividad que está vinculada con el mismo. Cada primitiva de servicio puede tener o no parámetros en los cuales se transmite la información requerida para proporcionar el servicio.

Las primitivas se clasifican en los siguientes cuatro tipos generales:

- **Request** (Petición): esta primitiva se utiliza para pedir el inicio de un servicio.
- **Confirm** (Confirmación): esta primitiva se utiliza para enviar los resultados de una o más peticiones anteriores de servicio asociadas.
- **Indication** (Indicación): esta primitiva se utiliza para proporcionar a la capa superior información sobre un evento interno.
- **Response** (Respuesta): esta primitiva se utiliza para completar un procedimiento previamente invocado por una primitiva de indicación.

### 3.3.6 Seguridad

Desde una perspectiva de seguridad, las VPANs definidas en el estándar presentan algunas diferencias con respecto a otras redes inalámbricas debido a la direccionalidad y visibilidad establecida como consecuencia de la elección del espectro óptico visible. Gracias a la direccionalidad y la visibilidad, si un receptor no autorizado se encuentra

en el camino de la señal de comunicación puede ser reconocido. Además, la señal no viajará a través de medios como paredes, a diferencia de otras redes inalámbricas basadas en radiofrecuencia, por ello siempre se conocerá a los dispositivos que estén comunicándose. Sin embargo, el estándar también proporciona algoritmos de seguridad para asegurar características como la confidencialidad de datos, autenticación y protección contra la repetición.

Los dispositivos pueden ser de bajo coste y por tanto tener capacidades limitadas en términos de potencia de computación, almacenamiento disponible y fuga de potencia, y no se puede asegurar que dispongan de una base de computación de confianza o un generador de números aleatorios de alta capacidad. Estas restricciones limitan la elección de protocolos y algoritmos de cifrado, e influye en el diseño de la arquitectura de seguridad debido a la necesidad del establecimiento de relaciones de confianza entre dispositivos cuyo direccionamiento debe realizarse con cuidado. Muchos de los elementos de la arquitectura de seguridad necesarios pueden ser implementados en capas superiores y por tanto se consideran fuera de los objetivos del estándar.

El mecanismo de cifrado que se va a presentar está basado en el cifrado por clave simétrica y utiliza claves que son proporcionadas por procesos establecidos en capas superiores. Este mecanismo asegura una aplicación segura de las operaciones de cifrado y un almacenamiento seguro de la información de cifrado mediante el uso de medios de autenticación.

El mecanismo de cifrado proporciona combinaciones particulares de los siguientes servicios de seguridad:

- **Confidencialidad de datos:** seguridad de que la información transmitida solo se muestra a las partes para las que está destinada.
- **Autenticidad de datos:** seguridad en el conocimiento de la fuente de la información transmitida y, por tanto, de que la información no ha sido modificada en el tránsito al receptor.
- **Protección contra repetición:** seguridad de que se detecta la información duplicada.

La protección de trama puede ser adaptada trama por trama y permite aplicar diversidad de niveles en la autenticidad de los datos (para minimizar la seguridad en las tramas transmitidas cuando sea requerido).

En la protección de la trama se permite el uso de una clave compartida entre dos dispositivos pares (clave de enlace) o compartida entre un grupo de dispositivos (clave de grupo), de este modo se permite cierta flexibilidad. Si una clave de grupo se utiliza para comunicaciones punto a punto, la protección se proporciona solo contra dispositivos externos y no contra potenciales dispositivos maliciosos en el grupo al que se ha compartido la clave.

### 3.4 La capa MAC

La capa MAC del estándar IEEE 802.15.7 es responsable de diversas tareas, entre las que se destacan las siguientes: gestión de balizas, acceso al canal, gestión de la ranura de tiempo garantizado (GTS), validación de trama, envío de trama de asentimiento, asociación y des-asociación. La subcapa MAC proporciona la base para la implementación de mecanismos de seguridad apropiados para cada aplicación. Otras características que proporciona esta subcapa son la función de color, visibilidad, estabilización de color y soporte para el control de atenuación.

Describiremos las características y funciones de la capa MAC del estándar IEEE802.15.7 relacionadas directamente con los objetivos del trabajo realizado en este proyecto.

### 3.4.1 Estructura de la supertrama

La capa MAC permite el uso opcional de una estructura temporal denominada supertrama, cuyas características vienen definidas por el coordinador. La supertrama es un patrón temporal que viene delimitado por tramas baliza (*beacons*). El coordinador transmite las tramas baliza al principio de cada supertrama para sincronizar los dispositivos, identificar la VPAN y definir las características de la supertrama.

La supertrama puede tener una parte activa y una inactiva. La parte activa está dividida en 16 intervalos de tiempo o *slots* equiespaciados, durante los cuales se realizan todas las comunicaciones. La parte inactiva permite que los dispositivos pasen al estado de ahorro de energía.

La parte activa se divide en un periodo de acceso con contención o CAP (*Contention Access Period*), en el que los dispositivos compiten entre sí para acceder al canal empleando un procedimiento de acceso aleatorio, y un periodo de acceso sin contención o CFP (*Contention Free Period*), destinado a aplicaciones con requisitos de latencia y ancho de banda. En el CFP el acceso al medio se realiza de forma centralizada mediante el uso de *slots* denominados GTS (*Guaranteed Time Slot*), los cuales son asignados por el coordinador.

La estructura de la supertrama viene definida por varios parámetros y atributos de configuración de la MAC PIB (*MAC Layer Personal Area Network Information Base*) que se describen a continuación:

- El atributo *macBeaconOrder*, *BO*, define el intervalo usado por el coordinador para transmitir sus tramas baliza. Los atributos *BO* y *aBaseSuperframeDuration* definen la duración – en ciclos de reloj – del intervalo de tiempo entre tramas baliza, *BI* (*beacon interval*), de acuerdo a la expresión (3.1):

$$BI = aBaseSuperframeDuration \times 2BO \quad \text{para: } 0 \leq BO \leq 14 \quad (3.1)$$

Si  $BO = 15$  el coordinador no transmite tramas baliza.

donde *aBaseSuperframeDuration* se define como el número de ciclos de reloj que forman la supertrama cuando el atributo *macSuperframeOrder* (véase a continuación) es igual a 0.

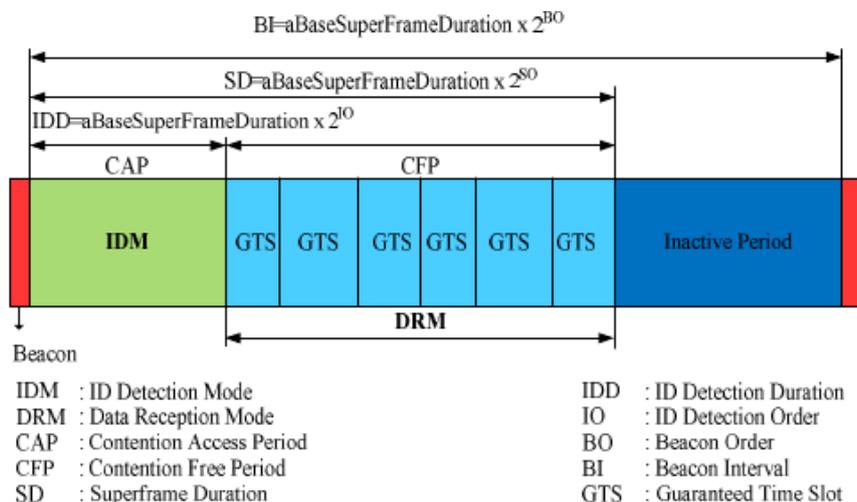
- El atributo *macSuperframeOrder*, *SO*, define la longitud de la parte activa de la supertrama, incluyendo la trama baliza. La relación entre el valor de *SO* y la duración de la parte activa de la supertrama, *SD* (*superframe duration*), medida en ciclos de reloj viene dada por (3.2):

$$SD = aBaseSuperframeDuration \times 2SO \quad \text{para: } 0 \leq SO \leq BO \leq 14 \quad (3.2)$$

Si  $SO = 15$  la trama no tendrá parte activa.

La trama baliza se transmite al comienzo del primer slot de la supertrama sin hacer uso de ningún método de acceso aleatorio. El CAP comienza inmediatamente después de la trama baliza, y el CFP – si existe – comienza a continuación del CAP, extendiéndose hasta el final de la parte activa de la supertrama.

En la **Figura 3.4** se ilustra un ejemplo de configuración de supertrama, en la que el intervalo de baliza BI tiene el doble de duración que la parte activa de la supertrama SD, y el CFP contiene dos GTS.



**Figura 3.4.** Estructura de supertrama MAC.

En el caso de que el coordinador no desee el uso de la estructura de supertrama no realizará transmisiones de balizas. Las balizas sirven para sincronizar los dispositivos conectados, para identificar la VPAN y para describir la estructura de supertrama a utilizar. Cualquier dispositivo que vaya a conectarse durante el CAP entre dos balizas, disputa el acceso al canal con otros dispositivos a través del acceso aleatorio ranurado.

Para aplicaciones de baja latencia o aquellas que requieran un ancho de banda de datos específico, el coordinador puede dedicar partes de la supertrama activa para esa aplicación. A esa parte se le llama slots de tiempo garantizado (GTSs) y el coordinador puede asignar un número de esos GTSs. Así mismo un GTS puede ocupar más de un periodo de slot, más adelante se proporcionará más información sobre esto.

Todas las transacciones con procesos de contención deben haberse completado antes de que empiece el periodo libre de contención (CFP). Además, cada dispositivo que transmita en un GTS debe asegurarse de que esa transacción se ha completado antes de que comience el próximo GTS o el final del CFP.

### 3.4.2 Control de acceso al medio: acceso aleatorio

Las VPAN del estándar IEEE 802.15.7 emplean varios mecanismos para mejorar la probabilidad de éxito en la transmisión de datos como el acceso aleatorio, el asentimiento de trama y la verificación de datos.

El acceso al medio regula el acceso al medio óptico de transmisión, permitiéndose efectuar transmisiones programadas o realizadas mediante métodos de acceso al medio aleatorios – con contención. Se definen cuatro métodos de acceso aleatorio: acceso aleatorio no ranurado, acceso aleatorio ranurado, CSMA/CA no ranurado y CSMA/CA ranurado.

Cuando se utiliza la supertrama en una VPAN, para transmitir en el CAP la capa MAC utiliza la versión ranurada del algoritmo de acceso aleatorio definido en el estándar. Si no se usan tramas baliza no hay supertrama definida y se emplea la versión no ranurada del algoritmo.

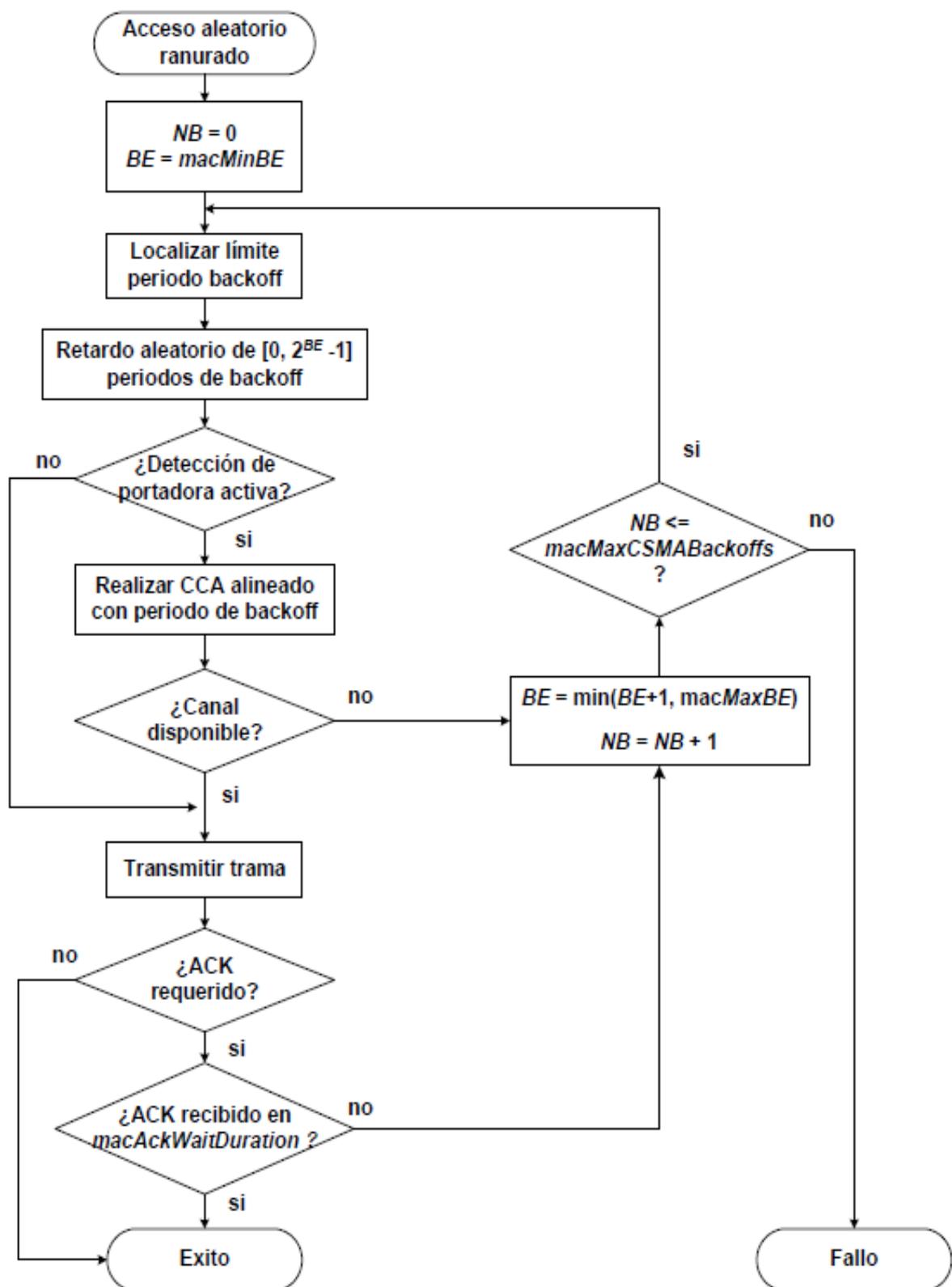


Figura 3.5. Versión ranurada del algoritmo de acceso aleatorio.

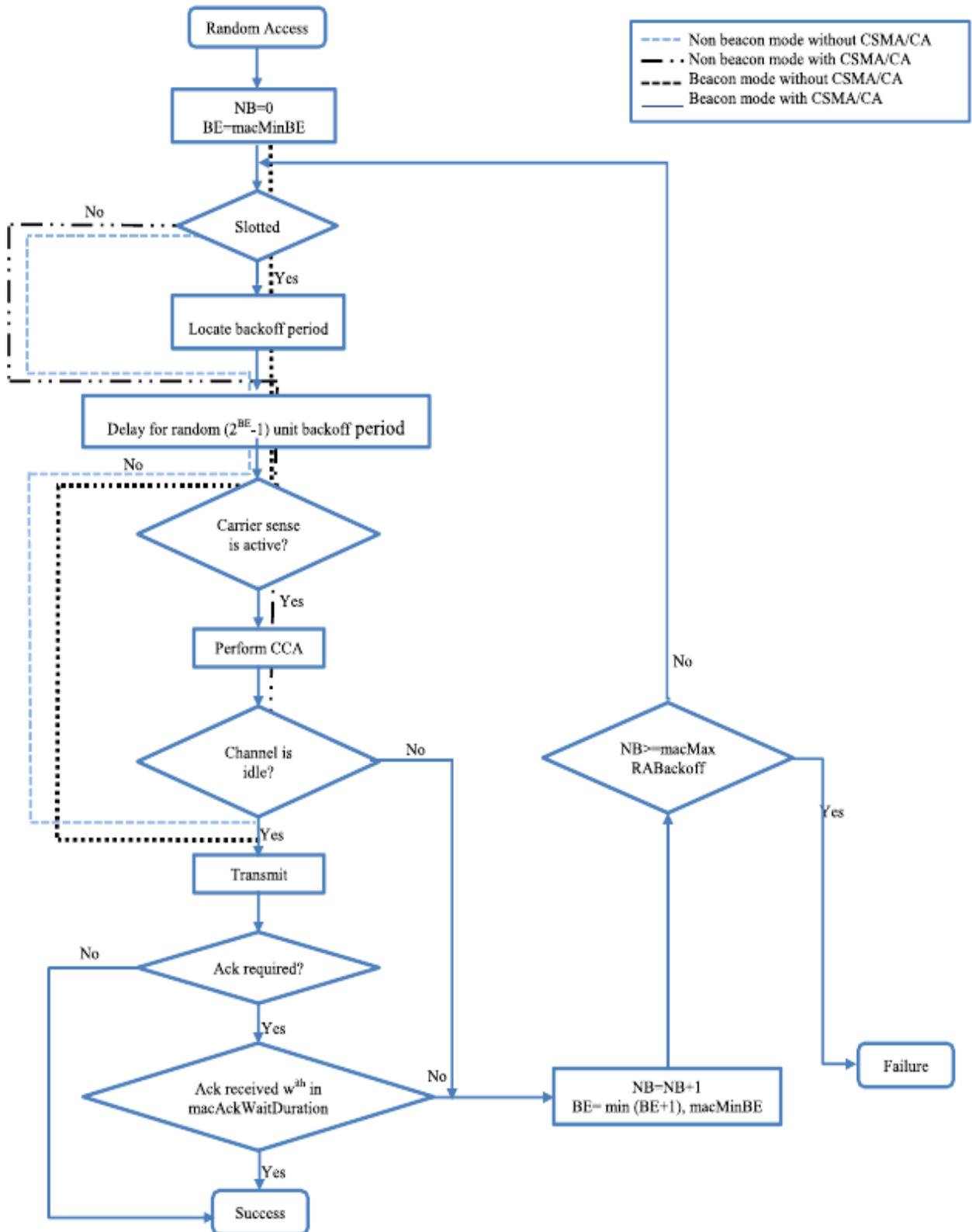


Figura 3.6. Diagrama de flujo del mecanismo de acceso aleatorio para todos los accesos.

En ambos casos el algoritmo de acceso se implementa mediante unidades de tiempo denominadas periodos de *backoff* (cuenta atrás), siendo la duración de un periodo de *backoff* igual al parámetro *aUnitBackoffPeriod*, expresado en ciclos de reloj:

$$aUnitBackoffPeriod = 20 \text{ ciclos de reloj} \quad (3.3)$$

Cuando se utiliza el acceso aleatorio ranurado, los periodos de *backoff* de los dispositivos de la VPAN deben estar alineados con los slots de la supertrama, de forma que el comienzo del primer periodo de *backoff* de cada dispositivo debe coincidir con el comienzo de la transmisión de la trama baliza.

En la **Figura 3.5** se muestra el diagrama de flujo de la versión ranurada del algoritmo de acceso aleatorio empleado en IEEE 802.15.7, que es descrito a continuación. En la **Figura 3.6** se describe el algoritmo de acceso aleatorio implementado y objeto de este PFC, en la capa MAC, incluyendo las cuatro variantes posibles.

Durante cada intento de transmisión de una trama realizado mediante el algoritmo de acceso aleatorio, cada dispositivo mantiene dos variables:

- **NB (Number of Backoffs)**: variable que contabiliza el número de veces que el algoritmo de acceso al medio ha realizado un *backoff* – un retardo aleatorio mediante una cuenta atrás – durante el actual intento de transmisión de la trama. Esta variable se inicializa a cero al comenzar el procedimiento de transmisión de una trama.
- **BE (Backoff Exponent)**: variable que guarda relación con el número de periodos de *backoff* que esperará el dispositivo antes de comprobar el estado del canal o acceder al mismo para transmitir. Esta variable se inicializa al valor de atributo *macMinBE*.

El procedimiento de acceso comienza con la inicialización de las variables *NB* y *BE*. A continuación, por usarse el acceso aleatorio ranurado, es necesario alinear el comienzo de los periodos de *backoff* del dispositivo con el comienzo de la supertrama.

Seguidamente se realiza el retardo de un número entero de periodos de *backoff* escogido aleatoriamente en el rango  $[0, 2BE - 1]$ . Si el número de periodos de *backoff* del retardo es mayor que el número de periodos de *backoff* que restan en el CAP de la supertrama, la capa MAC interrumpe la cuenta atrás al final del CAP y continúa al comienzo del CAP de la siguiente supertrama.

Una vez finalizado el *backoff*, la capa MAC continuará el procedimiento de acceso si el resto de los pasos del algoritmo, incluyendo la transmisión de la trama – y en su caso la correspondiente confirmación – pueden realizarse antes de que termine el CAP. En caso contrario, esperará al comienzo del CAP de la siguiente supertrama y realizará un nuevo retardo de *backoff*.

En el caso de que la capa MAC pueda continuar el procedimiento, y si está habilitada la opción CCA (*Clear Channel Assessment*), se solicita a la capa PHY que realice la función CCA para comprobar si el canal está disponible o hay transmisiones en curso.

Si la capa PHY evalúa que el canal está ocupado, la capa MAC incrementará las variables *NB* y *BE*, asegurándose de que *BE* no sea mayor que *macMaxBE*. Si *NB* es menor o igual al valor del atributo *macMaxCSMABackoffs*, se comenzará un nuevo retardo de *backoff*.

Si el valor de  $NB$  es mayor que  $macMaxCSMABackoffs$ , el algoritmo terminará con un fallo de acceso al canal, al haberse sobrepasado el máximo número de backoffs permitidos para intentar transmitir la trama. Si la capa PHY evalúa que el canal está disponible – o si la opción CCA no está activa – se realizará la transmisión de la trama. En la versión ranurada del algoritmo de acceso aleatorio tanto la operación de CCA como la transmisión de la trama deben estar alineadas con el comienzo de un slot de backoff.

Después de transmitir la trama, si está habilitado el uso de tramas de confirmación o ACK (*Acknowledgement*), se esperará por la trama ACK procedente del receptor hasta un tiempo máximo, dado por el atributo  $macAckWaitDuration$ . En el caso de recibirse la trama ACK se dará por concluido el procedimiento con éxito.

En el caso de no recibirse la trama ACK se procederá a incrementar las variables  $NB$  y  $BE$  para realizar un nuevo backoff siempre que  $NB$  no sea mayor que  $macMaxCSMABackoff$ .

La versión ranurada del algoritmo de acceso al medio empleado en IEEE 802.15.7 es muy similar a la empleada en el estándar IEEE 802.15.4, aunque hay que destacar algunas diferencias importantes:

- En IEEE 802.15.7 se invoca una sola vez la función CCA para comprobar si el canal está disponible antes de transmitir. En IEEE 802.15.4, cuando se emplea el mecanismo de acceso ranurado, se comprueba la disponibilidad del canal invocando la función CCA dos veces.
- En IEEE 802.15.7 tanto la detección del canal ocupado como la no recepción de la trama de confirmación (ACK) hacen que se realice un nuevo retardo de backoff (si no se superan los límites máximos) incrementándose las variables  $NB$  y  $BE$ . En IEEE 802.15.4 la retransmisión de una trama por no recibirse su confirmación supone un nuevo comienzo del procedimiento de acceso, con la inicialización de las variables  $NB$  y  $BE$ .

El algoritmo de acceso aleatorio al canal de la capa MAC, en su versión ranurada o general, se emplea en todas las transmisiones de tramas de datos o comandos a excepción de los siguientes casos:

1. Tramas balizas (*beacon*) transmitidas en una red VPAN trabajando en modo ranurado.
2. Tramas de reconocimiento (ACK).
3. Tramas de datos en el período sin contención CFP; esto es, en un GTS asignado.
4. Tramas de datos que puedan ser transmitidas rápidamente tras el reconocimiento de un comando de solicitud de datos pendientes.

### 3.4.3 Formato general y tipos de tramas MAC

La capa MAC define un sencillo formato de unidad de datos de protocolo de nivel MAC (MPDU) y que permite la realización de todas las funcionalidades propias de esta capa, y a las que se las conoce con el nombre de tramas MAC. El formato general de las tramas MAC incluye los siguientes campos:

- Una cabecera *MHR* (*MAC Header*): que incluye un campo de control, el número de secuencia, los campos de direccionamiento, e información relacionada con los mecanismos de seguridad del MAC.

MHR						MSDU	MFR	
Frame Control (2 bytes)	Seq Number (1 byte)	Addressing Fields				Aux Security Header (0/5/6/10/ 14)	Payload (variable)	FCS
		Dest VPAN Id (0/2 byte)	Dest Address (0/2/8 bytes)	Src VPAN Id (0/2 byte)	Src Address (0/2/8 bytes)			

Figura 3.7. Formato general de las tramas MAC.

- Un campo de datos *MSDU* (*MAC Service Data Unit*): que contiene la carga útil (*payload*) de la trama, presente en aquellos tipos de trama que lo requieran. Cuando existe, contiene información que depende del tipo de trama y es de longitud variable.
- Un campo de finalización *MFR* (*MAC Footer*): que existe cuando la trama incluye un campo de MSDU. Contiene una secuencia de verificación de trama de tipo CRC.

Los campos de la cabecera MHR aparecen en un orden fijo, aunque algunos de ellos, como los campos de direccionamiento, no aparecen en todos los tipos de tramas. En la **Figura 3.7** se muestra el formato general de las tramas MAC.

El estándar define cinco tipos de tramas MAC:

1. **Tramas *baliza* (*beacon*)**: enviadas por el coordinador de la red. Sirven para sincronizar los dispositivos e identificar la VPAN, así como para delimitar y definir las características de la supertrama,
2. **Tramas de *datos***: contienen los datos de la capa superior que deben ser transmitidos por la capa MAC.
3. **Tramas de *confirmación o reconocimiento* (*ACK*)**: confirman la correcta recepción de una trama de datos.
4. **Tramas de *órdenes***: mediante las que realizan diversas funciones de gestión y control de la capa MAC. Algunos ejemplos de tipos de tramas de órdenes son: solicitud y respuesta de asociación/disociación, solicitud y respuesta de petición de slot GTS, notificación de cambio de frecuencia de reloj, notificación de conflicto de identificador de VPAN, etc.
5. **Tramas *CVD* (*Color Visibility and Dimming frame*)**: se utilizan para enviar – en su carga útil – patrones de visibilidad, que son patrones de inactividad dentro de banda. Este tipo de tramas se emplea para notificar al usuario dándole información visual del estado o la calidad de la comunicación (por ejemplo, para avisar de un mal alineamiento entre dispositivos).

Para mejorar la probabilidad de éxito en la transmisión se implementa, el asentimiento de trama y la verificación de datos:

- ***Asentimiento de trama***: el éxito en la recepción y validación de la trama de datos o instrucción MAC puede confirmarse con un asentimiento. Si el dispositivo receptor no está disponible para gestionar la trama de datos recibida por alguna razón, el mensaje no será asentido.

Si el dispositivo origen del mensaje no recibe el asentimiento después de un periodo de tiempo, se asumirá que la transmisión no se ha realizado con éxito. Cuando no se requiere el asentimiento, el dispositivo origen asumirá que la transmisión se ha realizado correctamente.

- ***Verificación de datos***: Se incluye control de redundancia cíclico en la trama MAC y en la cabecera PHY para verificar la validez de los datos recibidos.

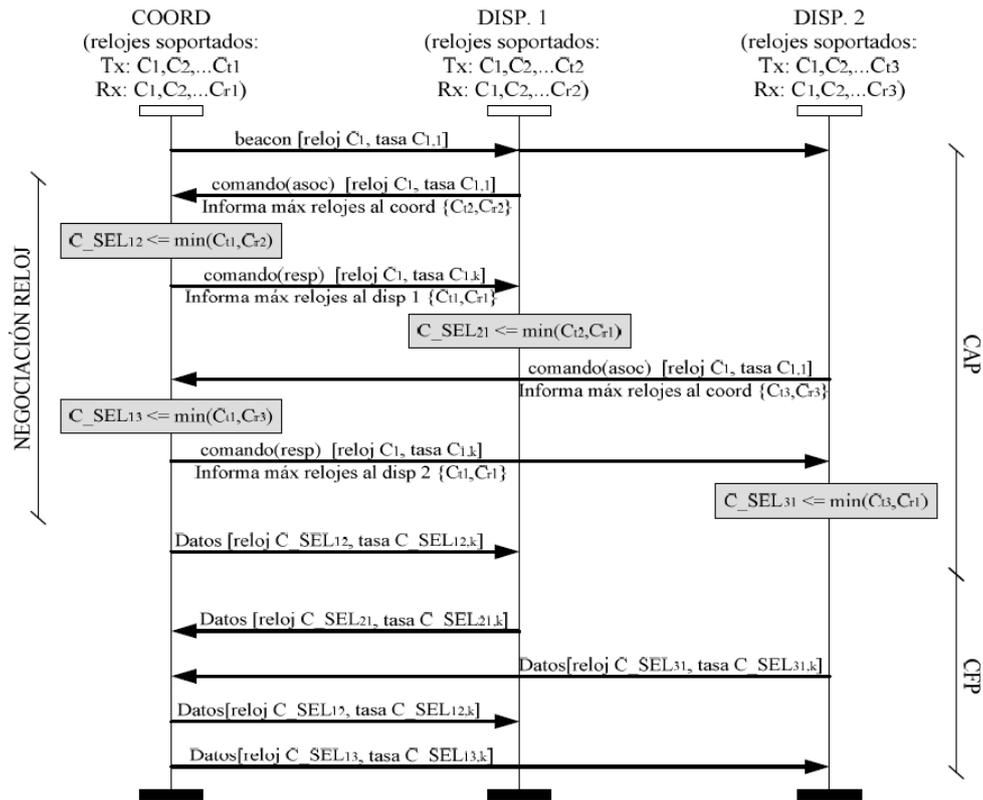


Figura 3.8. Negociación de reloj en la capa MAC.

### 3.4.4 Negociación del reloj en la capa MAC

El estándar IEEE802.15.7 soporta varias velocidades de reloj a fin de facilitar el empleo de un amplio abanico de transmisores y receptores ópticos. También admite el uso de velocidades de reloj asimétricas entre los caminos de ida y vuelta de un enlace óptico, posibilitado éste hecho por la separación espacial de los mismos. Por tanto, se requieren mecanismos de negociación de las velocidades de reloj a emplear entre dos dispositivos a comunicarse.

La negociación del reloj es realizada por la subcapa MAC y entre entidades pares durante la fase de descubrimiento de los dispositivos, donde todos los intercambios se producen a la mínima velocidad de reloj del modo PHY (modo obligatorio). En esa fase tiene lugar un intercambio de las capacidades de cada dispositivo en tramas comando cualesquiera, por ejemplo en las de asociación.

Entre esas capacidades están los modos PHY soportados, la máxima velocidad de reloj soportada y la característica de requerir la notificación explícita del reloj. Estas capacidades intercambiadas posibilitan, a las entidades pares MAC interactuando, seleccionar los relojes de trabajo permitidos en base a sus respectivos relojes máximos.

Realizada la selección de los relojes en ambas direcciones de la comunicación, la negociación de reloj puede terminar de dos formas: con notificación explícita del reloj previa a la comunicación o sin ella. La notificación explícita de reloj implica que los dispositivos transmitirán un tipo de trama comando específico de notificación de cambio de reloj antes de que la nueva velocidad sea usada.

Por contra, si no se requiere notificación explícita de reloj, los dispositivos tendrán la capacidad de detectar automáticamente el reloj de la trama de datos recibida, siempre

que sea menor o igual al máximo reloj soportado, y sin necesidad de ninguna notificación anterior.

Una secuencia temporal del mecanismo de negociación de reloj sin notificación explícita para una red en estrella (con varios dispositivos y con posibilidad de relojes asimétricos) se muestra en la **Figura 3.8**.

### 3.4.5 Constantes y atributos de la capa MAC

La entidad MLME (*MAC Sublayer Management Entity*) es la responsable de mantener la base de datos MAC PIB, que contiene las constantes y atributos gestionados por la capa MAC. En la **Tabla 3.2** se presentan las constantes de la capa MAC que tienen más relevancia en cuanto al funcionamiento del modelo de simulación implementado para este proyecto.

La MAC PIB contiene también los atributos necesarios para gestionar el funcionamiento de la capa MAC, incluyendo los atributos *macBeaconOrder* (*BO*) y *macSuperframeOrder* (*SO*), descritos en el apartado 3.4.1. En la **Tabla 3.3** se muestran los restantes atributos de la capa MAC que tienen especial importancia en el funcionamiento del modelo de simulación implementado para este proyecto.

**Tabla 3.2.** Principales constantes de la capa MAC (IEEE 802.15.7)

Constante	Descripción	Valor
<i>aBaseSlotDuration</i>	Número de ciclos de reloj que forman un slot de la supertrama cuando <i>macSuperframeOrder = 0</i>	60
<i>aBaseSuperframeDuration</i>	Número de ciclos de reloj que forman la supertrama cuando <i>macSuperframeOrder = 0</i>	$aBaseSlotDuration \times aNumSuperframeSlots$
<i>aNumSuperframeSlots</i>	Número de slots contenidos en la supertrama	16
<i>aUnitBackoffPeriod</i>	Número de ciclos de reloj que forman el periodo de <i>backoff</i> usado en el algoritmo de acceso aleatorio	20

**Tabla 3.3.** Principales atributos de la capa MAC (IEEE 802.15.7)

Atributo	Descripción	Rango valores	Valor defecto
<i>macAckWaitDuration</i>	Máximo número de ciclos de reloj a esperar por una trama ACK después de haber transmitido una trama de datos	Su valor se obtiene como la suma del tiempo necesario para comenzar la transmisión de la trama ACK y la duración de la misma	Depende del modo de trabajo del PHY seleccionado .
<i>macMaxBE</i>	Valor máximo de la variable <i>BE</i> (Backoff Exponent) usada en el algoritmo de acceso aleatorio	3 - 15	5
<i>macMinBE</i>	Valor mínimo de la variable <i>BE</i> (Backoff Exponent) usada en el algoritmo de acceso aleatorio	0 - <i>macMaxBE</i>	3
<i>macMaxCSMABackoffs</i>	Máximo número de retardos de backoff a realizar en el algoritmo de acceso aleatorio antes de declarar un fallo de acceso al canal	0 - 5	4
<i>macMaxFrameRetries</i>	Máximo número de reintentos de transmisión de una trama antes de declarar un fallo de transmisión de trama	0 - 7	3

### 3.4.6 Otras funciones de la capa MAC

Mencionamos aquí brevemente algunas funciones fundamentales relacionadas con el trabajo desarrollado en este PFC.

#### 3.4.6.1 Inicio y mantenimiento de una VPAN

Para los dispositivos pueden explorar el canal para comprobar el estado del mismo, detectar la emisión de tramas baliza (*beacons*) emitidas por coordinadores que estén en su rango de cobertura, o localizar una trama baliza en particular para sincronizarse con ella. El resultado de la exploración del canal permite saber qué canales lógicos (*channel number* o *Band Plan Id*) y/o identificadores de VPAN están disponibles, y cuáles están siendo usados por otras redes existentes.

#### 3.4.6.2 Asociación y disociación

La asociación permite a un dispositivo unirse a una red VPAN existente, y a un coordinador autorizar a un dispositivo para que se una a su VPAN.

La disociación finaliza la unión de un dispositivo a una VPAN, y puede ser iniciada tanto por el dispositivo como por el coordinador.

#### 3.4.6.3 Soporte de la función de color

Un dispositivo puede facilitar al usuario información de su estado haciendo que su señal óptica se visualice con un color determinado. La función de color puede, por ejemplo, servir de ayuda para que el usuario alinee adecuadamente el dispositivo para una correcta comunicación.

#### 3.4.6.4 Soporte para la iluminación y el control de la luminosidad

Incluye el uso de preámbulos extendidos para mejorar la sincronización, y el ajuste de la planificación de las transmisiones para adecuarse a un determinado nivel de brillo fijado para la iluminación.

La disociación finaliza la unión de un dispositivo a una VPAN, y puede ser iniciada tanto por el dispositivo como por el coordinador.

Como ya se ha dicho antes, el estándar ofrece soporte para funciones en el control de la visibilidad, control del nivel de iluminación o intensidad de la luz, control del parpadeo y en la estabilización del color. Todas estas funciones normalmente implican la inserción de elementos como patrones de visibilidad, tiempos de compensación o tramas de visibilidad durante los tiempos de inactividad en las comunicaciones. Las diferentes técnicas empleadas son altamente dependientes de los esquemas de modulación y codificación que se estén usando.

#### 3.4.6.5 Movilidad

En el caso de que se utilicen infraestructuras que incluyan múltiples transmisores y/o receptores ópticos para soportar la comunicación mediante celdas.

#### 3.4.6.6 Espaciado entre tramas (IFS)

La capa MAC necesita un periodo de tiempo para procesar los datos que ha recibido de la capa PHY. Para que esto sea posible, dos tramas sucesivas transmitidas desde un

dispositivo deberán estar separadas por al menos un periodo IFS. Si la primera transmisión requiere de asentimiento, la separación entre la trama de asentimiento y la segunda transmisión será de al menos un periodo IFS.

La duración del periodo IFS depende del tamaño de la trama que ha sido transmitida:

- Las tramas (por ejemplo, MPDUs) con longitudes menores o iguales que *aMaxSIFSFrameSize* octetos deberán seguirse por un periodo SIFS (espacio entre tramas corto) de duración como mínimo *macMinSIFSPeriod* relojes ópticos.
- Las tramas (por ejemplo, MPDUs) con longitudes mayores a *aMaxSIFSFrameSize* octetos deberán estar seguidas por un periodo de tiempo (LIFS, espacio entre tramas largo) de una duración de al menos *macMinLIFSPeriod* relojes ópticos.
- Las tramas de rotura (*burst*) tendrán un RIFS (espacio entre tramas reducido) de exactamente *macMinRIFSPeriod* relojes ópticos.

El atributo *aMaxSIFSFrameSize* de la subcapa MAC establece el límite entre las tramas consideradas largas y las consideradas cortas. Su valor es de 18 bytes. las tres posibles duraciones del IFS en ciclos de reloj: largo (LIFS) de 400 ciclos, corto (SIFS) de 120 ciclos, o reducido (RIFS) de 40 ciclos.

El algoritmo de acceso aleatorio ranurado deberá tener en consideración estos requerimientos para transmisiones en el CAP.

### 3.5. Comparación de IEEE-802.15.7 y IEEE-802.15.4

El estándar IEEE-802.15.7, desarrollado por el IEEE, para redes inalámbricas de área personal usando luz visible presenta muchas similitudes con otro estándar anterior desarrollado por el mismo organismo. Ese estándar es el IEEE-802.15.4 [76], también definido para el entorno de las redes inalámbricas de área personal, pero destinado a redes de sensores de radio con bajas velocidades de datos, y en el que prima la baja complejidad de los dispositivos y el ahorro de energía.

Ambas normas presentan especificaciones similares en lo referente a las funciones de comunicación, estando sólo las diferencias más notables en el hecho de que IEEE802.15.7 presenta funciones relacionadas con la función complementaria de iluminación que, obviamente, no aparecen en IEEE-802.15.4. Por su lado, IEEE-802.15.4 se focaliza más en la reducción del coste de implementación y en el ahorro de energía, al tratarse de dispositivos típicamente alimentados por baterías.

Sus arquitecturas son iguales, definiendo ambas normas las especificaciones de las capas de control de acceso al medio (MAC) y físicas (PHY). Las capas superiores esperadas en ambos casos también son iguales, si bien el estándar IEEE-802.15.7 admite la presencia de una entidad de gestión de la luminosidad (DME).

Las capas MAC de ambos estándares presentan un mayor parecido, si cabe, que la capa PHY. Esto es así pues las funciones de comunicación especificadas en las capas MAC de ambos estándares son iguales, hecho permitido por la abstracción del medio físico y del tipo de señal. La capa PHY presenta más elementos diferenciales entre ambas normas, como cabe esperar al emplear señales portadoras de diferente naturaleza (ópticas vs radio). Aun así, las especificaciones de las respectivas capas PHY de ambos estándares siguen filosofías similares

### 3.6. Conclusiones

En este capítulo se analizan las principales características de la capa MAC del estándar IEEE802.15.7. La capa MAC administra todo el acceso a la capa física PHY y es la responsable de un gran conjunto de tareas. Con una sola estructura de trama MAC se proporcionan capacidades punto a punto, en estrella o en difusión. Todos estos modos son soportados a través de una estructura de trama integrada de baja complejidad.

Las constantes y atributos que están especificados y mantenidos por la subcapa MAC se escriben en cursiva en el texto de este apartado. Las constantes tienen un prefijo general “*a*” y los atributos “*mac*”.

Todas las funciones y servicios ofrecidos por la subcapa MAC son accesibles a través de los dos puntos de acceso a servicios: MCPS-SAP y MLME-SAP. Tanto los servicios de datos como los de gestión se materializan a través de un conjunto de primitivas. El documento del estándar IEEE802.15.7 recoge el comportamiento que implica cada primitiva existente, así como los parámetros de cada una de ellas. En el **Anexo I** se proporciona en detalle el conjunto de primitivas de la capa MAC.



# Capítulo 4

## Herramientas de simulación: *OMNeT++*

---

*Si tu única herramienta es un martillo, tiendes a tratar cada problema como si fuera un clavo.*

ABRAHAM MASLOW, (1908-1970)

### 4.1. Introducción

Para el modelado y estudio de redes, es común usar herramientas de simulación que permitan representar escenarios y situaciones reales, así como reproducir el envío y recepción de tramas de datos entre dispositivos, conexión y desconexión, colisiones, posibles problemas, distintos protocolos, etc.

En este capítulo se presentan las diversas opciones existentes en el contexto de herramientas de simulación de red que permitirían llevar a cabo la implementación de la capa MAC de la norma IEEE 802.15.7/VLC [5]. En primer lugar, como estudio previo al desarrollo del objetivo de este proyecto, se ha llevado a cabo un análisis de la herramienta de simulación para analizar aquellas opciones que pudieran ser consideradas. Se repasan las principales características de dichas herramientas de simulación de red indicando su validez o no para la realización del objetivo de este proyecto fin de carrera.

Entre las alternativas vigentes, se encuentra el simulador abierto OMNeT++ [6], que ofrece una serie de ventajas frente a otros. Así, a continuación del estudio del arte, se introduce la plataforma de simulación OMNeT++ y el marco de trabajo INETMAMET escogidos finalmente para implementar la capa MAC de la norma IEEE 802.15.7/VLC. Posteriormente, profundizaremos en las propiedades que se adecuan a las características del entorno de trabajo donde se desarrolla el proyecto.

### 4.2. Descripción del estado del arte en simulación de red

El estándar IEEE 802.15.7, aunque ya no reciente, cuenta con escasos proveedores de soluciones software que lo implementen o incluyan como una componente estándar dentro de sus herramientas de simulación de red. En la mayoría de los casos, los módulos disponibles en las herramientas software que implementan la norma IEEE 802.15.7/VLC son fruto de contribuciones externas de carácter Open Source y, casi siempre, se encuentran en fase de desarrollo sin proporcionar demasiadas garantías de un correcto funcionamiento. Como consecuencia de este grado de inmadurez, es escasa la documentación y el soporte técnico correspondiente a dichos módulos y así el desarrollador se encuentra con la necesidad de crear sus propios módulos o adaptar los ya existentes.

Con el fin de ilustrar el contexto en el cual nos encontramos, a continuación nombraremos algunas de las herramientas de simulación de red más relevantes que existen en la actualidad.

### 4.2.1. NS-2 (Network Simulator 2)

Software de carácter libre implementado para la simulación de redes basado en eventos discretos, que surgió a finales de 1980 y cuya base es el simulador de redes REAL. Se trata de uno de los simuladores de redes gratuitos más ampliamente utilizado entre la comunidad docente e investigadora del área de redes de computadores, originalmente desarrollado dentro del proyecto VINT. NS-2 está desarrollado en C++ y su interfaz en Otcl [77].

NS-2 tiene capacidad para simular tanto protocolos *unicast* como *multicast*, con mayor uso en la investigación de redes móviles ad-hoc, también soporta una gran variedad de protocolos tanto en redes cableadas como en redes *wireless*.

Permite simular tanto protocolos de la capa de enlace como CSMA/CD, protocolos y algoritmos de encaminamiento, de transporte como TCP y RTP, *multicast*, y de la capa de aplicación como HTTP, TELNET y FTP. Además, también permite simular el nivel de enlace de redes 802.11.

NS-2 permite modelar redes terrestres, inalámbricas y por satélite con varios algoritmos de encaminamiento (DV, LS, PIM-DM, PIM-SM, AODV, DSR), distintas fuentes de tráfico (Web, ftp, telnet, cbr), e incluir fallos como pérdidas probabilísticas y deterministas, o fallos en la conexión. Nos permite visualizar: flujo del paquete, su encolado y su posible descarte; comportamientos del protocolo, como el comienzo lento de TCP, control de congestión, retransmisión rápida y recuperación; el movimiento de nodos en redes inalámbricas; y los estados del protocolo simulado.

La versión actual, NS-3, está diseñada para soportar todo el flujo de trabajo de la simulación desde la configuración hasta la recolección y análisis de tramas. Sin embargo, NS no es un producto terminado en ninguna de sus versiones. En particular, se han descubierto muchos bugs en el código y, además, en su mayor parte el código no está verificado por ninguna entidad verificadora, por lo tanto, ofrece una gran incertidumbre en los resultados que entrega.

### 4.2.2. OPNET Modeler

Es un programa ampliamente utilizado en la industria para modelar y simular sistemas de comunicaciones. Permite diseñar y estudiar redes, dispositivos, protocolos y aplicaciones, brindando escalabilidad y flexibilidad, cualidades que le permiten ofrecer a sus usuarios trabajar en procesos de investigación y desarrollo [78].

Esta herramienta proporciona un entorno virtual de red que modela el comportamiento de una red por completo, incluyendo sus pasarelas (*routers*), conmutadores (*switches*), protocolos, servidores y aplicaciones en red. Este entorno de trabajo es de gran utilidad ya que permite diagnosticar problemas de una forma eficiente, validar cambios en la red antes de implementarlos y prever el comportamiento de la red ante futuros escenarios como crecimiento de tráfico, fallos de red, etc.

El nombre corresponde a las siglas de *Optimized Network Engineering Tool*. Este software fue desarrollado en el año 1984 en el Instituto de Tecnología de Massachusetts (MIT) por investigadores y científicos del Laboratorio de Información y Decisión de Sistemas (LIDS). Está basado en la teoría de redes de colas e incorpora las librerías para facilitar el modelado de las topologías de red. Soporta un amplio rango de tecnologías de red tipo LAN, MAN y WAN.

Proporciona acceso directo al código fuente siendo esto una gran ventaja para desarrollar módulos nuevos. Opnet Modeler es una poderosa herramienta que permite simular sistemas de comunicaciones y así evaluar las prestaciones de una red bajo diversas condiciones de simulación como: flujos variables de tráfico, pérdida de paquetes o de conexiones entre terminales y la estación base, caídas de enlaces, etc.

El módulo de OPNET “Aplicación para la Caracterización del Entorno” (ApplicationCharacterization Environment, ACE) permite identificar de raíz problemas existentes en las prestaciones de las aplicaciones en red, y resolver estos problemas de manera eficiente y con bajo coste. Además permite cuantificar el impacto de los cambios realizados.

Ventajas y desventajas de OPNET son:

- Dispone de miles de modelos para todo tipo de equipos y enlaces
- Grado de realismo muy alto.
- Es lento simulando.
- Para muchas de nuestras aplicaciones, resulta excesivo y difícil de manejar.
- Flexibilidad media.
- Diseñado para simular casos prácticos/concretos.
- Más difícil de usar para simulaciones teóricas/generales.

### 4.2.3. GNS3

GNS3 es un simulador muy potente que permite, mediante un entorno gráfico, dibujar y configurar una topología de red y posteriormente simular su comportamiento. Soporta configuración y emulación de dispositivos de interconexión, *routers*, con sistema operativo IOS CISCO, también permite incorporar hosts (máquinas Linux o Windows) a través de *VirtualBox* a la topología de red diseñada [79].

Este software permite simular niveles de enlace diversos como *Ethernet*, *Frame Relay*, *ATM*, etc., así como dispositivos de interconexión del nivel de enlace como *switch*. Además, el tráfico que se genera en la red simulada, puede ser capturado con el software de monitorización de paquetes *Wireshark*.

Entre los inconvenientes de esta herramienta destacan la necesidad de tener buenos recursos de memoria RAM, y que el programa no trae consigo las imágenes IOS de los equipos/*routers* que pretende simular. Resulta más orientada, y una herramienta complementaria a los verdaderos laboratorios para los administradores de redes de Cisco o las personas que quieren pasar los exámenes CCNA, CCNP, CCIE DAC o certificaciones. Además es una herramienta orientada a la simulación de redes que al estudio y simulación de protocolos. Por tanto, no resulta de interés para la realización de este proyecto.

### 4.2.4. CNET Network Simulator

Es CNET es un simulador que permite experimentar y simular paquetes de datos en las capas de enlace, red y transporte en redes LAN (Ethernet IEEE 802.3) [80]. Así, si se quiere estudiar el direccionamiento, la detección de colisiones o el enrutamiento en función de un peso de transmisión asignado a cada enlace de redes LAN compuestas por varios segmentos de datos con tecnología Ethernet 802.3 unidas a través de *routers*, CNET es una herramienta muy interesante desde un punto de vista didáctico.

Además, puede ser interesante para la simulación prestacional de nodos y puntos de acceso de redes WLAN (IEEE 802.11) que utilizan el protocolo de acceso al medio

CSMA/CA. CNET está programado en lenguaje C, y puede ser ejecutado en sistemas operativos Linux, UNIX, OS-X o Mac y se distribuye bajo licencia pública GNU (GPL). Ha sido desarrollado por el departamento de Ciencias de la Computación e Ingeniería de la Universidad Western Australia.

La simulación de Ethernet es razonable, pero no perfecta. CNET soporta una velocidad de transmisión fija de 10Mbps, un tiempo de slot fijo 52  $\mu$ s, direccionamiento *broadcast*, detección de colisión, y bloqueo, pero no soporta el control de *jitter* ni el direccionamiento *multicast*. Las grandes redes deben construirse uniendo múltiples segmentos LAN a través de nodos *gateway* y enlaces WAN.

#### 4.2.5. J-Sim

Es J-Sim (antes Java Sim) no es propiamente un simulador de redes, se trata de una librería orientada a objetos para cualquier tipo de simulación de procesos discretos [81]. El motor de simulación de J-Sim y su entorno y objetos está programado en Java. J-Sim es ejecutable en sistemas operativos Windows, Linux y Unix, siempre que se disponga de la versión Java 1.5 o superior para su ejecución.

J-Sim dispone de paquetes y clases para simular y emular redes de sensores inalámbricos y los protocolos de la capa física y enlace de una red IEEE 802.11.

Se distribuye bajo licencia Academic v.2.1 y OSI Certified Open Source Software. Ha sido desarrollado por el departamento de Ciencias de la Computación e Ingeniería de la Universidad West Bohemia de la República Checa. La última versión disponible es la v.0.6.0 que data de Agosto de 2006, con lo cual no cuenta con soporte y no resulta fiable para nuevos desarrollos.

#### 4.2.6. VisualSense

VisualSense [82] es una plataforma específica para redes de sensores desarrollada dentro del proyecto Ptolemy de la universidad de Berkeley. VisualSense es una plataforma de código abierto para el modelado visual de sistemas basados en redes inalámbricas y de sensores. Está implementado sobre PtolemyII, el entorno software para el modelado, simulación y diseño sistemas embebidos, concurrentes, en tiempo real, del proyecto Ptolemy. Se trata de un proyecto sostenido a lo largo de los años (el primer Ptolemy II, data del año 2000 y el primer Ptolemy del año 93), VisualSense está disponible como parte de PtolemyII desde 2004.

Proporciona modelos básicos que puede ser modificados utilizando Java, se pueden combinar los modelos provistos para crear nuevos modelos complejos en la GUI (interfaz gráfica de usuario), o crear modelos nuevos partiendo de las “clases padre” proporcionadas en el entorno. También tiene la capacidad de combinar los modelos con comportamientos definidas en distintos dominios (con tiempo continuo, de eventos discretos, de flujo de datos síncronos, máquinas de estados finitos, sistemas inalámbricos, etc.).

La plataforma es muy potente ya que dentro de un mismo escenario permite tener cuantos modelos diferentes queramos, integrar diferentes canales, añadir obstáculos, etc. Cuenta con una interfaz de usuario relativamente sencilla y fácil de manejar permitiendo gran flexibilidad en cuanto a los tipos de datos que maneja, si bien presenta el gran inconveniente de no contar con bloques de alto nivel ya implementados. En general

proporciona modelos y clases con funcionalidades básicas para que el usuario las amplíe y/o combine para formar modelos más complejos.

En el estado actual de desarrollo de los modelos sería muy difícil modelar más allá de funcionalidades básicas, como saber el camino que seguirán los mensajes de un punto a otro o hacer una estimación con bastante margen de error de la vida útil del nodo.

Esta herramienta requiere, por tanto, mejorar la precisión de los modelos, y completar el módulo de comunicaciones exigiendo un gran esfuerzo de desarrollo inicial.

#### 4.2.7. OMNEST

Este simulador de eventos discretos es cada vez más popular entre la comunidad científica. Su área principal de aplicación es la simulación de redes de comunicaciones y el análisis y evaluación de éstas. Sus virtudes son múltiples. Los modelos implementados se basan en componentes que se hallan programados en C++.

Al tratarse de código libre, el programador puede añadir librerías o modificar los componentes según sus necesidades, etc. En muchos sentidos OMNEST es similar a OPNET, con la ventaja de ser código libre.

El entorno de simulación es un entorno gráfico, amigable y que permite la modificación de parámetros durante el tiempo de simulación. La ejecución del código es muy veloz, máxime comparada con la de NS-2. La documentación existente es notable, con una importante comunidad de usuarios como soporte.

Al poseer una estructura modular no es necesario un conocimiento demasiado profundo de todos los módulos involucrados en una simulación, solamente se requiere conocer cómo se comunican entre ellos, proceso que se realiza mediante mensajes cuyo formato es muy simple. OMNEST se ejecuta en Linux, Mac OS X, Unix y Windows. Además de ser este software libre para uso académico, sin ánimo de lucro, también tiene su versión comercial.

También, destacar que tiene una amplia comunidad activa de programación y que su últimas versiones contienen modelos para simular protocolos como PPP, Ethernet, IP, TCP, UDP, Mobile IPv6, 802.11., etc. Su versión comercial se puede descargar en <http://www.omnest.com>.

#### 4.2.8. OMNeT++ (OMNeT 2008)

El simulador abierto OMNeT++ [80], que ofrece una serie de ventajas frente a otros. Está compuesto de un entorno de desarrollo basado en Eclipse, con lenguaje de base C++ y un lenguaje propio de red denominado NED, un entorno de ejecución gráfica y una serie de herramientas adicionales. Al ser modular y extensible, permite la incorporación de forma sencilla de librerías de simulaciones y marcos de trabajo adicionales, desarrollados como proyectos independientes. Además, OMNeT++ es libre para uso académico, sin fines de lucro, ampliamente utilizada en la comunidad científica mundial y en continuo desarrollo.

Para el control y visualización de resultados, el software permite dos entornos de ejecución, uno gráfico (*Tkenv*) que se muestra en la **Figura 4.1**, y otro a través de un intérprete de comandos (*Cmdenv*). Trabajando en modo gráfico, se permite una simulación paso a paso (de mensajes), hasta un determinado instante o simulación de forma continua con tres posibles velocidades (normal, rápida y exprés), lo que lo dota de mucha versatilidad. El intérprete de comandos es más veloz que el gráfico.

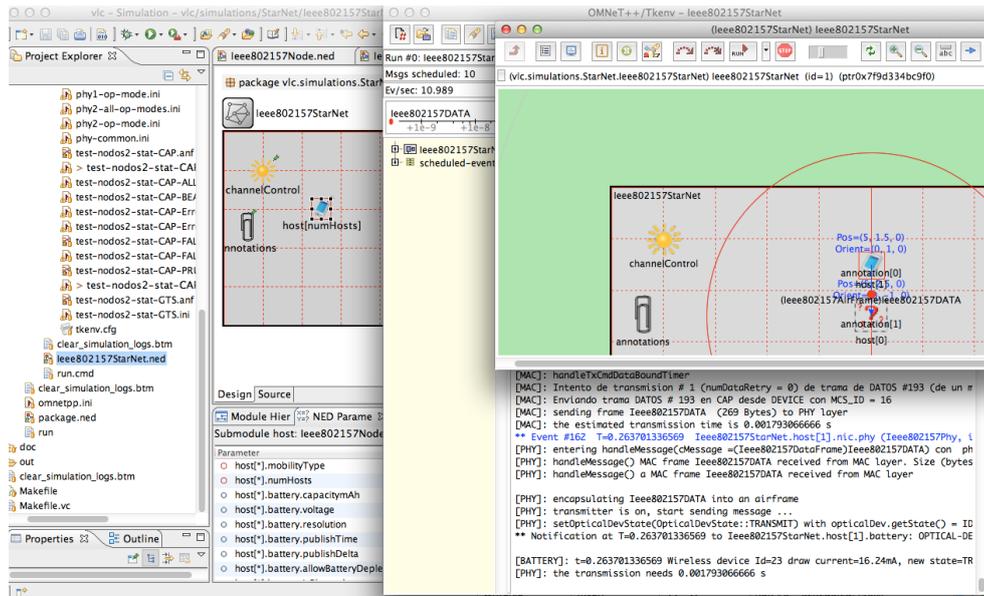


Figura 4.1. Entorno gráfico *Tkenv*.

Durante las primeras etapas de desarrollo la interfaz visual es una herramienta didáctica y de detección de errores muy útil aunque, conforme se avanza en el proceso, suele ser preferible el trabajo en modo comando al ser este último más veloz y eficiente para la recopilación de resultados y datos estadísticos.

Una de las ventajas que presenta esta herramienta es que está en pleno desarrollo, permite la implementación de protocolos y su estudio tanto en entornos de redes cableadas como inalámbricas. Por tanto muchos proyectos y protocolos están siendo desarrollados e incorporados en este entorno. Por lo que convertía a esta herramienta en candidata para desarrollar el objetivo de este proyecto fin carrera.

Existían dos motivos más para elegir esta herramienta como la herramienta de trabajo. Por un lado se contaba con una librería implementada con el protocolo IEEE 802.15.4, protocolo con características similares al IEEE 802.15.7. Y por otro lado en un proyecto anterior a este PFC, se desarrolló el nivel físico del mismo estándar, y que ha sido el cimiento para la simulación de la capa MAC. Se utilizó el marco de trabajo INETMANET para la implementación de dicha norma. Por tanto, como consecuencia de la relación entre todas estas condiciones y teniendo en consideración el tipo de red a simular, el trabajo a desarrollar, y las posibilidades de la herramientas, se ha optó por la continuidad del OMNeT++ como software para el estudio del presente PFC.

En el siguiente apartado se hará una introducción a OMNeT++.

### 4.3. Entorno de trabajo en OMNeT++

Como ya hemos comentado en el apartado anterior, OMNeT++ es un entorno de simulación de eventos discretos. Proporciona un conjunto de herramientas y componentes programados en C++ y cuya interfaz gráfica está basada en la plataforma Eclipse. Además, los distintos módulos programados en C++ se agrupan como objetos de alto nivel mediante un lenguaje de descripción de topología denominado NED, cuyo formato es muy sencillo y que refuerza el carácter modular del entorno. De este modo, su arquitectura modular que separa núcleo de simulación, modelos, interfaz gráfico, etc, permite fácilmente integrarlo en aplicaciones personalizadas.

Este entorno permite la ejecución interactiva de la simulación siendo de gran utilidad en las primeras fases de desarrollo como se ha comentado con anterioridad. Algunas de sus características principales son:

- Presentación de los resultados de la simulación durante el tiempo de ejecución.
- Ejecución paso a paso
- Visualización de los mensajes programados en una ventana independiente
- Examen y modificación de objetos y variables mediante las ventanas de inspección.
- Ventanas independientes para la salida de cada módulo.
- Breakpoints etiquetados.

Por otra parte, el entorno de línea de comandos *Cmdenv*, aunque menos intuitivo, dispone de un modo exprés, cuya ejecución es aún más veloz que la de su homónimo en *Tkenv*, motivo por el que resulta muy interesante una vez que el código se encuentra depurado y lo que prima es obtener los resultados de las simulaciones en el menor tiempo posible. Adicionalmente, es posible concatenar diversas simulaciones, cada una de ellas con sus parámetros correspondientes, y ejecutarlas secuencialmente, es decir, realizar un procesado por lotes (modo *batch*) mediante un fichero de configuración de OMNeT++, aislando por completo la simulación del sistema operativo sobre el que se esté ejecutando.

El modelado mediante eventos discretos en OMNeT++ consiste en una serie de módulos jerárquicamente anidados, los cuales se comunican entre sí mediante mensajes que se definen a partir de los ficheros de mensajes *.msg*. Para la definición de la estructura de los distintos elementos que conforman la red, y que se comunican con intercambio de paquetes, existen los ficheros de red *.ned*. Ambos tipos de ficheros son necesarios para obtener la simulación del modelo y la obtención de datos de salida.

### 4.3.1 Ficheros *.msg*

Permiten la definición de mensajes. Se pueden definir varios tipos de mensajes diferentes y agregar campos de datos a los mismos. OMNeT++ traducirá las definiciones de mensajes a clases C++ completamente desarrolladas.

Para el manejo y definición de los distintos tipos de eventos, OMNeT++ utiliza la herencia de clases. Todos los eventos, pertenecen a clases que heredan directa o indirectamente de una clase predefinida, llamada *cMessage*. Mediante un lenguaje descriptivo, se indica la estructura del mensaje, que previamente a la compilación será analizado, generando el código necesario. Estos ficheros de descripción, permiten la inclusión de diferentes códigos, para conseguir una elevada personalización del resultado.

En el siguiente ejemplo (**Código 4.1** y **Código 4.2**) podremos ver cómo de forma muy sencilla se van creando estructuras de mensajes heredadas (*Message*), y a partir de las cuales se genera el código C++ que las implementa (*Head c++*).

**Código 4.1.** Ejemplo de estructura de mensaje *Command Frame*.

```
message Ieee802157CmdFrame extends Ieee802157Frame  ///  
// Se generan eventos con  
// el envio de tramas Comando  
{  
    int cmdType enum(Ieee802157MacCmdType);        ///  
// Command Frame Identifier  
// Selecciona el tipo de trama  
// Comando  
    CapabilityIE cmdPayload;                        ///  
// Permite incluir el IE  
}
```

Código 4.2. Ejemplo de código de cabecera generado automáticamente para *Command Frame*.

```

/*
 * Class generated from <tt>Ieee802157Frame.msg</tt> by opp\_msgc.
 * <pre>
 * message Ieee802157CmdFrame extends Ieee802157Frame
 * {
 *   intcmdTypeEnum(Ieee802157MacCmdType);
 *   CapabilityIEcmdPayload;
 * }
 * </pre>
 */

class Ieee802157CmdFrame : public ::Ieee802157Frame
{
protected:
    int cmdType\_var;
    CapabilityIE cmdPayload\_var;

private:
    void copy(const Ieee802157CmdFrame& other);

protected:
    // protected and unimplemented operator==((), to prevent accidental usage
    bool operator==(const Ieee802157CmdFrame&);

public:
    Ieee802157CmdFrame(const char *name=NULL, int kind=0);
    Ieee802157CmdFrame(const Ieee802157CmdFrame& other);
    virtual ~Ieee802157CmdFrame();
    Ieee802157CmdFrame& operator=(const Ieee802157CmdFrame& other);
    virtual Ieee802157CmdFrame *dup() const {return new Ieee802157CmdFrame(*
    this);}
    virtual void parsimPack(cCommBuffer *b);
    virtual void parsimUnpack(cCommBuffer *b);

    // field getter/setter methods
    virtual int getCmdType() const;
    virtual void setCmdType(int cmdType);
    virtual CapabilityIE& getCmdPayload();
    virtual const CapabilityIE& getCmdPayload() const {return const\_cast<
    Ieee802157CmdFrame*>(this)->getCmdPayload();}
    virtual void setCmdPayload(const CapabilityIE& cmdPayload);
};

inline void doPacking(cCommBuffer *b, Ieee802157CmdFrame& obj) {obj.parsimPack(
b);}
inline void doUnpacking(cCommBuffer *b, Ieee802157CmdFrame& obj) {obj.
parsimUnpack(b);}

```

### 4.3.2 Ficheros *.ned*

Un fichero *.ned* describe un objeto mediante la lista de sus componentes. Para cada uno de los componentes se asigna un nombre, y es posible dar valores a sus parámetros.

Siguiendo este método de abstracción, una red de telecomunicaciones se describirá mediante un fichero *.ned*. En él se indicarán los distintos módulos, o equipos que la componen, y las conexiones que existen entre ellos. Cada uno de estos módulos podrá estar compuesto a su vez por otros, hasta llegar a un módulo simple. Este módulo simple no contendrá ningún otro módulo, y se corresponderá con un objeto de una clase previamente definida en un fichero C++ del mismo nombre. Es en este fichero de código fuente C++ donde se describe el funcionamiento del módulo simple, que

heredará directa o indirectamente de la clase *cModule* predefinida en las librerías de OMNeT++.

En el ejemplo **Código 4.3** pueden verse los distintos parámetros (*parameters*), puertas (*gates*) y conexiones (*connections*) que existen entre los componentes o módulos del fichero *.ned* de la **Figura 4.2**.

**Código 4.3.** Código fuente del fichero *.ned*. correspondiente al módulo de la **Figura 4.2**

```

module Ieee802157Node
{
  parameters:
    @node();
    string mobilityType;           // define el patron de movilidad del nodo (VER
                                  // en carpeta <tt>mobility</tt>)
    volatile int numHosts;         // debe estar de acuerdo con el de
    Ieee802157StarNet

  gates:
    input opticalIn;              // puerta optica de entrada al nodo. Es
                                  // inalambrica. No conectada a ningun otro nodo

  submodules:

    // submodulo de gestion de notificaciones
    notificationBoard: Ieee802157NotificationBoard
    {
      parameters:
        @display("p=65,70;i=block/control");
    }

    // submodulo de movilidad. Elegido por parametro. Debe obedecer a
    // modulo interfaz IMobility
    mobility: <mobilityType> like IMobility
    {
      parameters:
        @display("p=65,150;i=block/cogwheel");
    }

    battery: Ieee802157SimpleBattery // submodulo de bateria
    {
      parameters:
        @display("p=65,310;i=block/plug");
    }

    app: Ieee802157TestApp           // submodulo de aplicacion
    {
      parameters:
        @display("p=250,100;i=block/source");
    }

    net: Ieee802157StarRouting       // submodulo de red
    {
      parameters:
        @display("p=250,200;i=block/network2");
    }

    nic: Ieee802157Nic              // submodulo de interfaz de red (IEEE 802.15.7)
    {
      parameters:
        @display("p=250,300;q=queue;i=block/ifcard");
    }

    connections allowunconnected:
      opticalIn --> nic.opticalIn;
      nic.upperLayerOut --> net.lowerLayerIn;
      nic.upperLayerIn <-- net.lowerLayerOut;
      net.upperLayerOut --> app.lowerLayerIn;
      net.upperLayerIn <-- app.lowerLayerOut;
    }
}

```

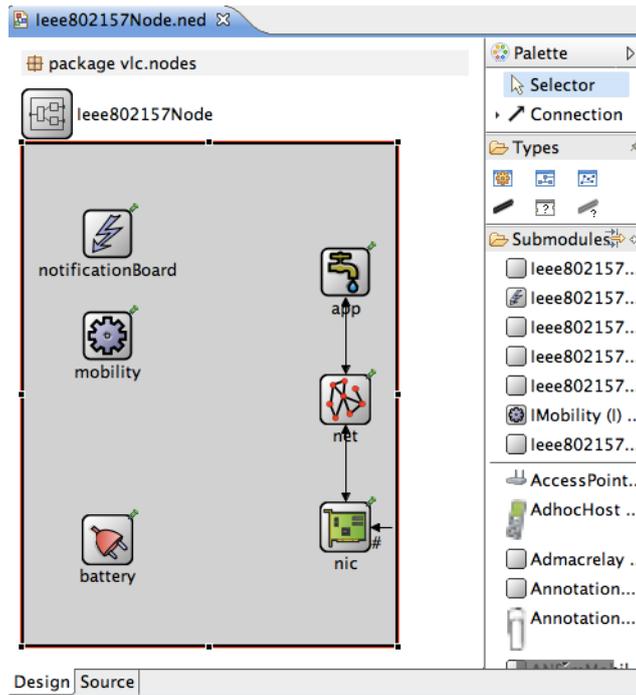


Figura 4.2. Diseño fichero *.ned*

### 4.3.3 Simulación con OMNeT++

Los programas de simulación se desarrollan a partir de los componentes *.msg* y *.ned*. Una simulación con OMNET++ requiere de tres fases: construcción, configuración y por último ejecución. Se describe a continuación cada una de ellas:

1. **Construcción:** en primer lugar, los ficheros *.msg* son traducidos a código C++ usando el programa *opp\_msgc*. Los ficheros *.ned* deben pasarse a C++ con el compilador NED. Los ficheros NED se cargan dinámicamente en su forma original como texto cuando el programa de simulación comienza. Tras ello, todos los ficheros fuente C++ son compilados y vinculados con el *kernel* de simulación y una librería de interfaz de usuario para crear un ejecutable de simulación o una librería compartida. La compilación genera los ficheros objeto *.obj*, a los que se enlazarán los ficheros de librerías estáticas. Estos pasos darán lugar al programa de simulación y darán por terminada la tarea de construcción
2. **Configuración:** el siguiente paso es la configuración de la simulación, o lo que es lo mismo, la creación y edición del fichero *omnetpp.ini*. Este fichero contiene los valores que controlan cómo se ejecuta la simulación, valores para los parámetros del modelo, etc.
3. **Ejecución:** Una vez realizada la configuración se procede a la ejecución que dará lugar a los ficheros de salida. En lo que se refiere a la ejecución de las simulaciones y el análisis de los resultados obtenidos, cabe mencionar que una simulación puede ser compilada como un programa ejecutable independiente, por lo que puede ser ejecutada en otras máquinas (sin necesidad de que OMNeT++ esté presente) o puede ser creada como una librería compartida.

Cabe hacer una mención de la interfaz de usuario que proporciona OMNeT++. Su propósito es hacer visible para el usuario las características internas del modelo, permitirle controlar la ejecución de la simulación, así como darle la posibilidad de intervenir modificando variables u objetos del modelo. Todo esto es de enorme utilidad en la etapa de desarrollo y *debugging*.

#### 4.3.4 Configuración de simulaciones en OMNeT++

Al configurar una simulación de OMNeT++ y la salida de datos es preciso llevar a cabo especificaciones sobre tres elementos base: funciones estadísticas empleadas, tipos y tiempo de simulación, y grabación de resultados de salida.

Posteriormente, en el **Capítulo 4** se describe en detalle la configuración definida para la implementación de simulación de redes VLC.

##### 4.3.4.1 Configuración de las funciones estadísticas

Para las funciones estadísticas, OMNET++ incluye numerosas funciones que permiten trabajar con diferentes tipos de distribuciones de variables aleatorias. Se utilizan en los archivos *.ned* e *.ini* también, para explotar la herramienta de simulación y recoger resultados según los comportamientos de red definidos bajo distintos escenarios. Existen principalmente dos tipos, las continuas y las discretas, y abarcan distribuciones uniformes, exponenciales, normales, etc.

##### 4.3.4.2 Configuración del tiempo de simulación

En las aplicaciones de OMNET++ hay que diferenciar entre tiempo real y tiempo simulado:

- Cuando se refiere a tiempo real o a tiempo de CPU, se refiere al tiempo real que tarda un ordenador en ejecutar la aplicación.
- El tiempo simulado se refiere al tiempo que se simula en una ejecución.
- Para establecer un límite a la simulación se introduce en el fichero de configuración ambos valores.
- Cuando se pretende establecer eventos en determinados instantes de tiempo, el camino es utilizar el tipo *simtime\_t*. Este tipo, declarado como un tipo *double* en la librería se emplea para almacenar el valor de tiempo de simulación. Cuando se cree una variable de este tipo, se puede llamar a la función *simTime()* que devuelve el valor actual de tiempo de simulación en un tipo *simtime\_t*.

##### 4.3.4.3 Configuración de la grabación de resultados

Los resultados de la simulación se pueden grabar para luego ser analizados a posteriori. Los datos de salida principalmente pueden ser de dos tipos, escalares y vectoriales.

En el primero de los casos, se graba el valor, pero no el tiempo en el que se ha grabado, por lo que únicamente aparece el dato. En el caso de los datos vectoriales, se graba el valor y además la marca de tiempo. Los datos escalares son muy útiles para mostrar valores totales, o porcentajes. Los datos vectoriales son más útiles para ver gráficas de la evolución de la simulación en el tiempo.

Para ambos tipos de datos se generan unos ficheros de salida con extensión *.sca* y *.vec*, según el caso escalar y vectorial, respectivamente, y con un nombre definido en la

configuración. Los datos escalares se graban generalmente al final de la simulación, aunque existe la posibilidad de poder grabarlos en cualquier otro momento de la misma.

Lo habitual es que se graben desde la función *finish()* de cada módulo. Para ello se debe realizar una llamada a la función *recordScalar.recordScalar(<Nombre>, <valor>)*. En el caso de los vectoriales, para la grabación de los resultados de la simulación se emplea la clase *cOutVector*, que guarda los datos de tipo vector. Cada objeto de esta clase puede ser un atributo de un módulo, o como un objeto independiente.

Se debe tener en cuenta que si se crea un objeto local, este será destruido cuando salga de la función. Es decir si se crea un objeto de tipo *cOutVector* en la función de manipulación de mensajes, cuando la función finalice, también se destruirá el objeto, por lo que se perderán los datos.

Si la intención es usar un objeto *cOutVector* para guardar datos de un único módulo, lo mejor es añadirlo a la declaración de ese módulo e iniciarlo en la función *initialize()*. Para darle un nombre se debe llamar al método *setName* de la clase *cOutVector* y pasarle una cadena de caracteres.

Este es el nombre con el que se asociará en la visualización de resultados *cOutVector* *<nombreVector>*. Para escribir los valores lo que se debe hacer es llamar a la función *record* de la clase *cOutVector* de la forma siguiente: *vectorName.record(<value>)*. La clase guarda automáticamente la marca de tiempo, por lo que no es necesario que sea el usuario el que la introduzca.

### 4.3.5 Análisis de resultados

Por último para el análisis de resultados tendremos datos de salida escalares y vectoriales. Éstos se escriben en los ficheros con extensión *.sca*, y *.vec* respectivamente, y cuyo nombre se define en el fichero de inicialización. Para analizar los resultados que ha dado como salida la simulación emplearemos las herramientas OMNeT++Scalars y OMNeT++ Plove.

La interpretación de cada uno de los archivos de salida recogidos en cada fase de implementación del PFC se describe en el **Capítulo 6**.

En resumen, la plataforma OMNeT++ de simulación de redes de comunicación se puede usar en escenarios de cualquier sistema que permita su modelado mediante eventos discretos y pueda ser fácilmente esquematizado mediante elementos que se comuniquen con intercambio de paquetes. Por tanto, un modelo en OMNeT++ consiste en una serie de módulos jerárquicamente anidados, los cuales se comunican entre sí mediante mensajes.

OMNeT++ proporciona los recursos básicos para realizar simulaciones, pero por sí mismo no da soporte a la simulación de ningún tipo específico de red, tecnología o arquitectura de protocolos. La funcionalidad necesaria para la simulación de redes y protocolos específicos es aportada por modelos elaborados por la comunidad de desarrolladores. Muchos de estos modelos se presentan agrupados en paquetes de simulación, entre los que se pueden destacar INET e INETMANET.

Este último marco de trabajo, INETMANET, fue escogido para este proyecto, y en el siguiente apartado se detalla el motivo de su elección y la descripción de su contenido.

## 4.4 Marco de trabajo para OMNeT++

Para desarrollar un modelo de comunicación sobre una herramienta que ofrezca unos recursos de simulación de eventos discretos y una plataforma con decisión a todos los niveles, se necesita una infraestructura compleja, la cual tiene un coste de implementación muy elevado y totalmente fuera del alcance de este proyecto. Es por ello, que la manera más rentable es recurrir a librerías de modelos de protocolos compatibles y a la simulación de redes o protocolos.

Actualmente, el marco de trabajo INET ofrece un paquete de simulación de redes de comunicación de código abierto para el entorno de simulación OMNeT++. El marco de simulación INET hace uso del mismo modo de funcionamiento, es decir, módulos que se comunican entre sí mediante paso de mensajes. Contiene modelos para varios protocolos de redes cableadas e inalámbricas, como *UDP*, *TCP*, *SCTP*, *IP*, *IPv6*, *Ethernet*, *802.11*, *Point-to-Point Protocol* (PPP), *Open Shortest Path First* (OSPF), *Multiprotocol Label Switching* (MPLS), algunos modelos de la capa de aplicación, y muchos otros.

No obstante, la funcionalidad de INET en la que nos centraremos es en la posibilidad que ofrece de simulación de redes móviles e inalámbricas. Este paquete, que aporta la componente inalámbrica al escenario de simulación, se podrá adaptar y completar para obtener un marco de trabajo con el que se simulare redes bajo el protocolo definido por el estándar IEEE 802.15.7 para VLC.

#### 4.4.1 Descripción de INET

El marco de trabajo INET [7] es una librería que proporciona protocolos, agentes y otros modelos válidos para investigaciones y proyectos de estudios de redes de comunicaciones. INET es especialmente útil para diseño y validación de nuevos protocolos, o exploración de nuevos escenarios.

INET contiene modelos para Internet (*TCP*, *UDP*, *IPv4*, *IPv6*, *OSPF*, *BGP*, etc.), protocolos de capas de acceso cableado o inalámbrico (*Ethernet*, *PPP*, *IEEE802.11*, *IEEE802.15.4*, etc.), soporte para movilidad, protocolos MANET, Di DiffServ, MLPS con LPD y señalización RSVP-TE, multitud de modelos de aplicación, y muchos otros protocolos y componentes.

Otros marcos de trabajo de simulación llevan INET como base, y lo extienden en direcciones específicas, tales como redes vehiculares, redes superpuestas o entre pares, o LTE.

Al igual que OMNeT++, INET está construida sobre el concepto de módulos que se comunican mediante transmisión de mensajes. Los agentes y protocolos de red se representan por componentes, los cuales pueden combinarse libremente para formar dispositivos de red como *hosts*, *rotures*, *switches*, y otros. El usuario puede programar nuevos componentes, y los existentes han sido escritos de forma que resultan de fácil comprensión y modificación.

Los beneficios de INET también provienen de la infraestructura aportada por la herramienta OMNeT++. Más allá de hacer uso de los servicios prestados por el *kernel* y librerías de la herramienta (modelo de componentes, parametrización, registro de resultados, etc.), los modelos de interés pueden ser desarrollados, ensamblados, parametrizados, ejecutados y sus resultados evaluados desde las comodidades de la interfaz gráfica de simulación OMNeT++, o desde la opción de línea de comando.

Otra ventaja es que el marco de trabajo INET se mantiene constantemente actualizado para la comunidad usuaria por el equipo de desarrollo de OMNeT++, encontrándose

frecuentemente parches para solucionar problemas y nuevos modelos aportados por los miembros de la comunidad.

Dentro de los muchos marcos de trabajos existentes basados en INET, mantenidos cada uno por grupos de investigación o de desarrolladores independientes, encontramos INETMANET, una ramificación de INET con características experimentales relacionadas con las redes móviles y ad-hoc. Este marco de trabajo será la base para nuestra investigación y desarrollo de este PFC.

#### 4.4.2 Arquitectura de INETMANET

El marco de trabajo INET se construye sobre OMNeT++, y usa el mismo concepto que la herramienta: módulos comunicándose por transferencia de mensajes. Generalmente provee las mismas funcionalidades que INET, pero contiene protocolos adicionales y componentes que son especialmente útiles para el modelado de comunicaciones inalámbricas.

Con este marco ampliado dispondremos de modelos de propagación, protocolos de capa de enlace, protocolos de encaminamiento móvil, modelos de movilidad y de aplicación, y otros cambios y modelos, todos ellos necesarios para simular el estándar VLC que necesitamos implementar en un escenario real de transmisión inalámbrica.

Para implementar nuevos protocolos o modificar los existentes, se necesitan añadir el nuevo código bajo el directorio fuente y por tanto, regenerar los ficheros *makefile*. A continuación se presenta la estructura y directorios presentes en el marco de trabajo y cuáles de éstos estarán principalmente implicados en la ejecución de redes Virtual Local Area Network (VLAN) del estándar para Comunicaciones por luz visible.

Los directorios de fuente del marco de trabajo se disponen a lo largo de las capas OSI según la estructura de la **Figura 4.3a**. Bajo esta estructura, y haciendo uso de las librerías asociadas al proyecto INETMANET [83], tendremos la estructura para el desarrollo del estándar VLC según vemos en la **Figura 4.3b**.

Esta estructura, organizada según el sistema de niveles OSI, al igual que el marco INETMANET, permite una diferenciación del nivel MAC que desarrolla este trabajo, y del nivel físico desarrollado en un proyecto relacionado, con el mismo objetivo de simulación del nuevo estándar de comunicación inalámbrica de corto alcance.

Conceptualmente la estructura seguida por el modelo VLAN será el mostrado en la **Figura 4.4a**. Tal y como se ha comentado en la descripción del funcionamiento de la herramienta y los archivos de configuración, vemos en la **Figura 4.4b** como el archivo *Ieee802157Nic* contiene la estructura de la capa de enlace.

#### 4.4.3 Módulos y protocolos

Los protocolos son representados por módulos simples. Una interfaz externa de un módulo simple (puertas, conectores y parámetros) se describen en un fichero *.ned*, y la implementación está contenida en una clase C++ con el mismo nombre.

Estos módulos pueden combinarse libremente para formar dispositivos de red con el lenguaje NED (no requiere código C++ ni recompilación). Se puede encontrar variedad de modelos de acceso en los nodos, pero también puedes crear los tuyos propios para adaptarlo a tus escenarios de simulación particulares.

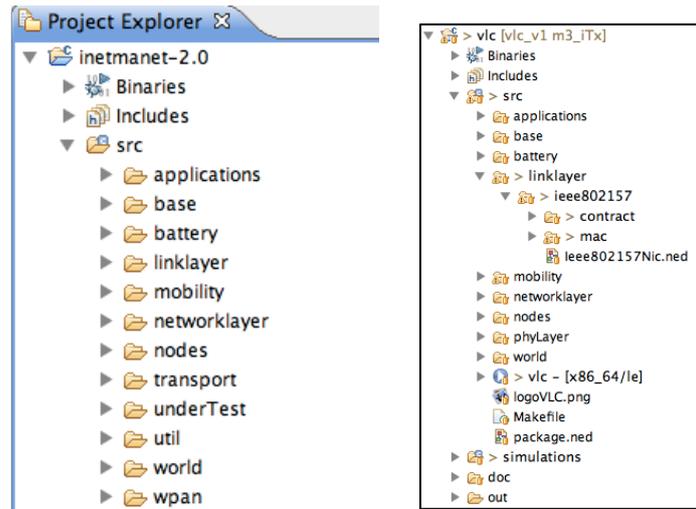


Figura 4.3. (a) Estructura de directorios INETMANET, vs. (b) Estructura de directorios VLC

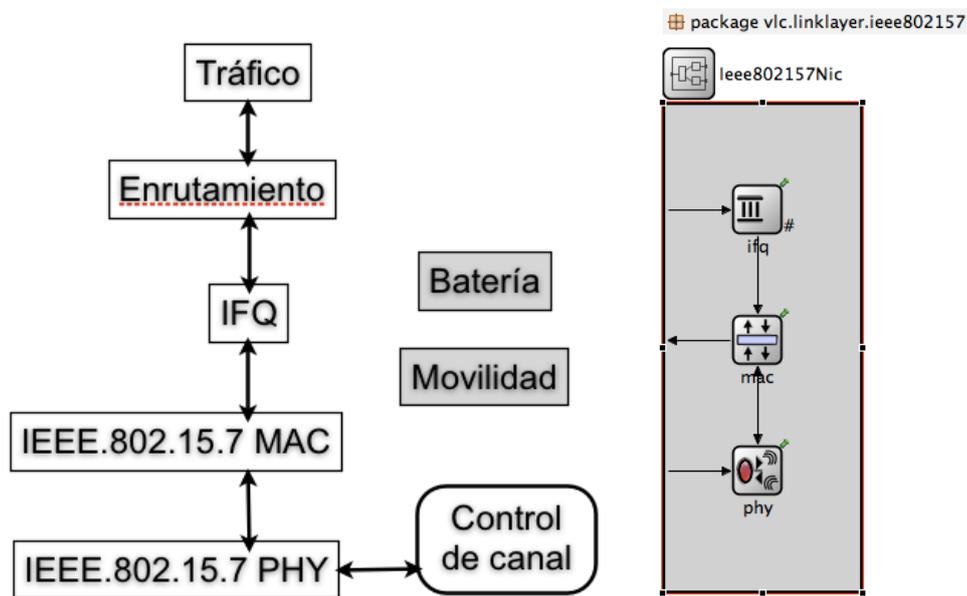


Figura 4.4. (a) Estructura conceptual del modelo IEEE802.15.7. (b) Estructura implementada de la capa de enlace del modelo IEEE802.15.7.

Las interfaces de red son así mismo, módulos compuestos (es decir, ensamblados de módulos simples), y son compuestos por colas, una MAC, y posiblemente otros módulos simples.

Sin embargo, no todos los módulos implementan protocolos. Hay módulos que poseen datos, facilitan comunicación de módulos, llevan a cabo la configuración automática de una red, se mueven alrededor de un nodo móvil, y realizan el servicio de barrido asociado a canales de radio en simulaciones inalámbricas.

Para el desarrollo software sobre INET de este PFC destacamos la importancia de los siguientes módulos: *NotificationBoard* y *ChannelControl*.

1. *NotificationBoard*: de entre los módulos comunes que aparecen en muchos host, router o dispositivos modelos, en este proyecto haremos estudio y uso del *NotificationBoard*. Este módulo permite a los diferentes módulos

comunicarse de forma publicación-suscripción. Esta forma consiste en difundir una notificación a todos los módulos suscritos a esa categoría de notificación, de un cambio de estado que la dispare. El submódulo de notificación en el host debe llamarse *notificationBoard* para que otros módulos lo encuentren.

Usando *NotificationBoard* los módulos pueden notificarse entre sí eventos como cambios de configuración en interfaces, transferencias inalámbricas, cambios en el estado del canal inalámbrico, cambios de posición del nodo móvil, etc. Tiene exactamente una instancia dentro de un modelo de dispositivo, y se accede a éste a través de llamadas de método C++ directas (no de intercambio de mensajes).

Los módulos pueden "suscribirse" a distintas categorías de cambios (por ejemplo, "tabla de enrutamiento cambió" o "canal de radio se convirtió en vacío"). Cuando se producen estos cambios, el módulo correspondiente (por ejemplo, el módulo de capa física) le permitirá al *NotificationBoard* saber, y se transmitirá esta información a todos los módulos interesadas.

En la **Tabla 4.1** se resumen las características principales de este nodo.

2. **ChannelControl:** a nivel global de red, están los módulos que sólo tienen una instancia. En concreto, para simulaciones de redes inalámbricas se requiere el *ChannelControl*. El módulo *ChannelControl* se informa acerca de la ubicación y el movimiento de los nodos, y determina qué nodos se encuentran dentro de la comunicación o la distancia de interferencia. Esta información es utilizada luego por las interfaces radioeléctricas de los nodos durante las transmisiones. Debe ser nombrado como *channelControl* dentro de la red.

En la **Tabla 4.2** se resumen las características principales de este nodo.

#### 4.4.4 Comunicación entre capas de protocolos

Los protocolos son representados por módulos simples. Una interfaz externa de un módulo En el marco de trabajo INET, cuando un protocolo de capa de nivel alto quiere enviar un paquete de datos a un protocolo de capa inferior, el módulo de capa superior sólo envía el objeto mensaje representando el paquete al módulo de capa inferior, el cual se volverá a encapsular y enviar. El proceso inverso se lleva a cabo cuando un protocolo de capa inferior recibe un paquete y lo lanza hacia arriba después de la desencapsulación.

A menudo es necesario transmitir información extra con el paquete. Esta información extra se adjunta al objeto mensaje como control *info*. *ControlInfo* son objetos pequeños de valores, los cuales son adjuntos a paquetes (objetos mensaje) con su función miembro *setControlInfo()*. *ControlInfo* sólo contienen información auxiliar para la siguiente capa de protocolo, y se supone que no debe ser enviado a través de la red a otros hosts.

**Tabla 4.1.** Módulo simple *NotificationBoard*

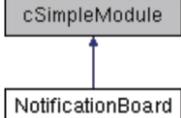
Módulo	Referencia de la clase	Descripción
 <b>NotificationBoard</b> Paquete: inet.base Fichero: src/world/radio/NotificationBoard.ned	Diagrama de herencia:  <pre> classDiagram     class cSimpleModule     class NotificationBoard     NotificationBoard -- &gt; cSimpleModule                     </pre>	Actúa como intermediario entre módulos donde pueden ocurrir cambios de estado, y módulos los cuales están interesados en conocer dichos cambios.

Tabla 4.2. Módulo simple *ChannelControl*

Módulo	Referencia de la clase	Descripción
 <i>Paquete:</i> inet.world.radio <i>Fichero:</i> src/world/radio/ChannelControl.ned	Diagrama de herencia: 	Este módulo de control lleva el seguimiento de cuáles nodos entran en la distancia de interferencia de otros nodos.

Las cabeceras de protocolos y formatos de paquete se describen en ficheros de definición de mensajes (ficheros *.msg*), los cuales son traducidos a clases C++ por la herramienta *opp\_msgc* de OMNeT++. Las clases de mensajes generadas son subclases de la clase *cMessage* de OMNeT++.

#### 4.5 Análisis de las funcionalidades de acceso al medio de INETMANET

Como se ha mencionado anteriormente, el entorno INETMANET contiene varios modelos de acceso al medio, entre ellos el del estándar IEEE.802.15.4. Sin embargo, para los objetivos de este proyecto es necesario implementar el modelo del nivel MAC de la norma IEEE 802.15.7.

Por tanto, una vez adquirido el conocimiento necesario sobre el software de simulación y las funciones del estándar requeridas, como el nivel de acceso al medio y sus mecanismos asociados, el siguiente paso ha sido estudiar con más detalle la viabilidad del código existente implementado en INETMANET respecto a los mecanismos MAC de VLC que se pretenden implementar en este PFC. A su vez, debemos analizar las diferencias generales entre ambos estándares de redes de área personal de corto alcance –norma IEEE 802.15.7 y norma IEEE 802.15.4.

INETMANET incluye las funciones y primitivas para el acceso de canal necesarias para la simulación de un modelo de WPAN para la transmisión y recepción de baliza completa para nodos formando automáticamente redes en topología estrella y *cluster tree*, incluyendo dos modos de transferencia de datos excluyentes entre sí –directo y GTS. Además aporta las funcionalidades de filtro de tramas MAC, detección de paquetes recibidos duplicados, y la parte más importante, el mecanismo de acceso al medio CSMA-CA (ranurado y no ranurado). Con el fin de hacer el modelo más exacto, INETMANET también considera el retardo IFS, el cual está definido como el tiempo de proceso requerido para una trama recibida de la capa PHY por la MAC.

Con respecto a las funciones de gestión de la PAN, está modelado un proceso de asociación simplificado, según el cual el nodo se asociará con un coordinador del cual recibe la primera baliza. La pérdida de sincronización debido a la deriva de tiempo puede ser simulado con este modelo.

Aunque INETMANET contiene suficientes mecanismos para gestión de WPAN, y no fue necesario modificarlos todos para el alcance de este proyecto, a continuación se resumen las similitudes y diferencias principales entre ambos estándares, y las carencias que se precisó cubrir.

1. **Similitudes:** Los protocolos para redes WPAN y VPAN tienen en común los siguientes comandos:

- Solicitud de asociación (implementado en su forma simple).
  - Respuesta de asociación.
  - Notificación de disociación.
  - Solicitud de dato (implementado).
  - Notificación de conflicto de PAN ID / VPAN ID (pequeña variación).
  - Solicitud de baliza.
  - Realineación de coordinador.
  - Solicitud de GTS (implementada su gestión).
2. ***Diferencias:*** se han identificado como principales diferencias entre ambos estándares las siguientes carencias:
- La exploración de canal y la disociación, no son considerados en el modelo de WPAN.
  - La diferencia del tiempo de ida y vuelta para el transceptor entre transmisión-recepción y recepción-transmisión es propia de VLC, por tanto carece de implementación en WPAN.
  - La regularización de tiempo sobre relojes ópticos en lugar de símbolos es una diferencia principal entre los protocolos, la cual carece por tanto de implementación.
  - La topología P2P en ambos estándares difieren, con lo que no se dispone de dicha configuración correctamente además de no disponer de la configuración para topología de difusión.
  - No existe la elección de velocidad de reloj.
  - No existen las tramas CVD para el control de visibilidad, que está dentro del alcance de este proyecto.
  - No existe el mecanismo CSMA sin evasión de colisiones, el cual sí está dentro del alcance de este proyecto.
  - No existe la trama de respuesta de asociación, con el campo correspondiente: *Capability\_Negotiation.Response*.
  - De entre los **comandos** que de los que carece el estándar IEEE.802.15.4, serán necesarios implementar:
    - Respuesta de GTS.
    - Notificación de cambio de tasa de reloj.
    - Inhabilitación de CVD.
    - Elemento individual de información (IE).
    - Recuperación rápida de conexión.
    - Notificación de movilidad.
    - Asignación de canal múltiple.
    - Notificación de parpadeo.
    - Notificación de control de intensidad.
    - Notificación de tiempo de la estabilización del color.
    - Información de la estabilización del color.

Se implementaron, por tanto, todas las primitivas, tramas y comandos necesarios para cubrir las carencias del modelo IEEE.802.15.4. La mayor parte de las funciones necesarias para el mecanismo de baliza habilitada están implementadas para la WPAN en INETMANET. Hubo que hacer la adaptación general del código para todas las funciones, donde incluía los métodos relacionados con dicho mecanismo, e incluir alguna funcionalidad más. Sin embargo, el algoritmo de acceso al canal conllevó

muchas mayores variaciones en el código y algunos métodos nuevos que se especifican y describen en el capítulo de implementación de esta memoria de proyecto.

## 4.6 Conclusiones

En este capítulo se analizan diversas opciones existentes en el contexto de herramientas de simulación de red que permitirían llevar a cabo la implementación de la capa MAC de la norma IEEE 802.15.7/VLC. Entre las alternativas vigentes, el simulador abierto OMNeT++, ofrece una serie de ventajas frente a otros. Se describe la plataforma de simulación OMNeT++ y el marco de trabajo INETMAMET escogidos finalmente para implementar la capa MAC de la norma IEEE 802.15.7/VLC.

Finalmente, se analizan las funcionalidades de acceso al medio que contiene INETMANET, puesto que este entorno sólo contiene el modelo MAC de la norma IEEE 802.15.4. Se detallan las carencias encontradas con respecto al estándar IEEE 802.15.7/VLC, y que son precisas implementar. El siguiente capítulo describe el conjunto de modificaciones y nuevos desarrollos llevados a cabo en este PFC.



# Capítulo 5

## Implementación de la capa MAC

---

*"Jamás se desvía uno tan lejos como cuando cree conocer el camino."*

PROVERBIO CHINO

### 5.1. Introducción

El objetivo primordial de este proyecto es la implementación del protocolo de acceso al medio que permitiese estudiar y analizar la transmisión de paquetes o tramas entre dispositivos usando el nivel de acceso al medio especificado por el estándar IEEE.802.15.7 para tecnología VLC. Después de un análisis exhaustivo tanto del escenario de desarrollo como del protocolo a desarrollar, llegamos a este punto dónde se explicará en lo posible, la implementación llevada a cabo, en C++ principalmente, para lograr dicho objetivo.

Para el desarrollo de las rutinas necesarias, básicamente, se adoptó parte de la estructura modular del modelo tomado como base y se aprovecharon funcionalidades programadas en sus módulos simples del PFC [9]. Las adaptaciones se centraron en el módulo asociado a la capa de acceso al medio o nivel MAC, donde se localizan la mayor parte de las tareas de construcción y programación del modelo, que añadieron nuevos elementos y funcionalidades y modificaron las existentes.

El alcance de este proyecto, abarca el desarrollo de tres mecanismos fundamentales para adaptar el módulo MAC del estándar 802.15.4 implementado en INETMANET de acuerdo a las especificaciones de la norma 802.15.7: el modo baliza, el acceso al medio CSMA/CA y la transferencia indirecta. Así mismo describiremos las dificultades encontradas en cada una de las funciones del estándar, las posibles soluciones planteadas, y el tratamiento del código final en cada caso.

Se describen, por tanto, cuatro apartados bien diferenciados para cada uno de los mecanismos o modos implementados en este PFC: el modo baliza, el acceso al medio CSMA/CA y la transferencia indirecta, y adaptación del código del módulo MAC 802.15.4 implementado en INETMANET. En los siguientes apartados veremos las soluciones desarrolladas y aquellas partes del código claves para la comprensión y justificación de dichas soluciones. Éstas se implementaron durante la etapa de construcción tal como se detalla a continuación, y en el mismo orden que exponemos en este capítulo

Con respecto a cada mecanismo, se describirá la situación inicial o punto de partida encontrada para plantear los cambios que convirtieron el nivel MAC implementado para WPAN en dicha capa de acceso al medio pero con el comportamiento propio de VLC (VPAN).

Se complementa este capítulo con tres anexos (**Anexo II**, **III** y **IV**) en los cuales se describen aspectos claves del estándar 802.11.7 para entender la implementación de los mecanismos involucrados.

## 5.2. Descripción funcional del modelo

El modelo de red VLC implementado está compuesto por una serie de unidades funcionales bien diferenciadas, como se deduce de la descripción funcional. Esas unidades funcionales, típicas de una arquitectura de comunicaciones, se materializan en módulos componentes, típicos de cualquier modelo de simulaciones de OMNET++. Esos módulos se organizan siguiendo una estructura modular jerárquica, la cual se define por el código NED presente en unos pocos archivos; básicamente en los tres archivos *.ned* que albergan los módulos compuestos presentes en el modelo:

- *Ieee802157StarNet*: módulo de red o modelo de simulaciones.
- *Ieee802157Node*: tipo de módulo compuesto representando un nodo.
- *Ieee802157Nic*: tipo de módulo compuesto representando la interfaz de red.

Esta actividad tuvo la dificultad añadida de no disponer de documentación adecuada y de existir una cantidad excesiva de líneas de código en una misma clase.

La estructura modular del modelo de red se completa con los tipos de módulos simples empleados en los niveles más bajos de la estructura modular jerárquica y con las definiciones de mensajes que representan a los diferentes tipos de paquetes de datos que se intercambian los módulos de la red.

El resultado de lo definido en dichos elementos de construcción del modelo (módulos compuestos, módulos simples y definiciones de mensajes) es la estructura modular completa. Como ejemplo, se muestra en la **Figura 5.1** los módulos componentes de cada nodo de la red.

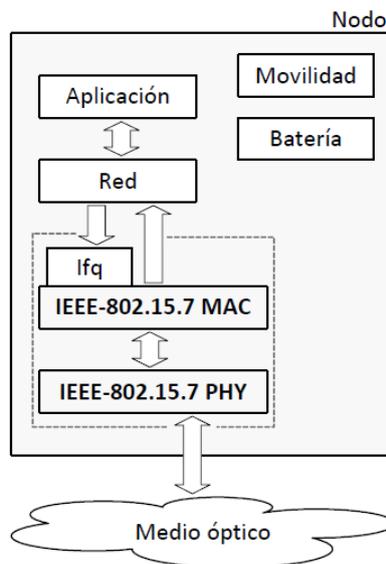


Figura 5.1. Módulos componentes de cada nodo de la red.

## 5.3. Metodología de trabajo

El procedimiento de trabajo seguido consistió en tomar como punto de partida el modelo de simulaciones del estándar IEEE-802.15.4 para OMNET++, incluido en el paquete de simulaciones INETMANET, y sobre éste realizar las modificaciones necesarias a fin de lograr el desarrollo del modelo de simulaciones de redes VLC que implementa las especificaciones de la capa física definidas en el estándar IEEE-802.15.7, también para OMNET++.

El acondicionamiento del código hace referencia a los cambios y soluciones aplicados al código existente del marco de trabajo INETMANET. Como origen nos encontramos un código, que aunque tiene muchas similitudes con el estándar de comunicación por luz visible, difiere de éste en nomenclatura, campos y tipos de tramas, y algunas funcionalidades.

La actividad inicial fue determinar el nivel de modificaciones necesario, conservando el funcionamiento de los mecanismos implementados y la respuesta frente a los escenarios de simulación, los cuales fueron creados para probar el nivel físico descrito en el proyecto desarrollado en paralelo a éste PFC [9]. Basándonos en las diferencias estudiadas en la fase previa del proyecto, y descritas en el capítulo anterior, se llevó a cabo esta fase de acondicionamiento, no tan compleja a nivel de codificación, pero que supuso un alto nivel de análisis de algoritmos ajenos, minuciosidad a la hora de llevar a cabo dichos cambios, y verificación continua.

Esta actividad tuvo la dificultad añadida de no disponer de documentación adecuada y de existir una cantidad excesiva de líneas de código en una misma clase. Para cada elemento integrado, modificado o eliminado se identificaron todas sus dependencias, elementos relacionados y se ejecutaron los escenarios de pruebas diseñados para verificar que no se veían afectadas las funcionalidades ya activas.

#### 5.4. Tareas de programación realizadas

Los objetivos a conseguir en este proyecto final de carrera es crear una versión del protocolo MAC que soporte los siguientes puntos:

1. Desarrollo de mecanismo de acceso múltiple por detección de portadora con y sin evasión de colisiones (CSMA y CSMA/CA) para mecanismo de acceso con baliza ranurada.
2. Modo con baliza habilitada.
3. Mecanismo de inicio de VPAN y mantenimiento.
4. Mecanismos de asociación y disociación. Comprobación de sincronización.
5. Modelo de transferencia directo y GTS (ranuras de tiempo garantizado).
  1. Control de espaciado entre tramas, filtrado y detección de duplicidad.
  6. Control de temporizaciones según la unidad de tiempo de relojes ópticos.
  7. Implementación de las primitivas necesarias para el funcionamiento de los puntos anteriores.
  8. Creación del formato de tramas para VLC, incluyendo las tramas CVD para mantener la intensidad de luz apropiada entre tramas de datos, soportar control de luminosidad, y para el suministro de información visual tales como el estado de la comunicación y la calidad del canal al usuario.
  9. Soporte para simulación de distintas propuestas de mecanismos que solventen el problema del control de evasión de colisiones, o también llamado problema del nodo oculto.
  10. Inclusión de parámetros estadísticos para la extracción de resultados tras la realización de simulaciones.

En los siguientes apartados, se describen los diferentes desarrollos alcanzados en este proyecto agrupados en cuatro bloques principales:

1. Acondicionamiento del código (**apartado 5.5**).
2. Modo baliza inhabilitada (**apartado 5.6**).
3. Modelo de acceso al medio CSMA/CA (**apartado 5.7**).
4. Modelo de transferencia indirecto (**apartado 5.8**).

## 5.5. Acondicionamiento del código

La adecuación del código (en nomenclatura, campos y tipos de tramas, y algunas funcionalidades) parte del estudio de viabilidad de WPAN (IEEE.802.15.4) de INETMANET para determinar el posible aprovechamiento de métodos en VPAN (IEEE.802.15.7) descrito en el apartado 4.5 y que concluía las principales carencias presentadas por el modelo MAC de la norma IEEE.802.15.4 implementado en INETMANET que deberían ser desarrolladas para usarse como módulo MAC de la norma IEEE.802.15.7. Los siguientes apartados recogen los cambios introducidos.

### 5.5.1. Estructura Task Pending

Esta estructura contiene como miembros el tipo de primitivas asociadas a la tarea pendiente (*task pending*), así como el paso de ejecución de dichas tareas, incluidas en el estándar IEEE.802.15.7, ambas inicializadas por defecto al valor false y 0 respectivamente. En la **Tabla 5.1**, consideramos el uso o no uso de dichos miembros en el código implementado, si también forman parte del estándar IEEE.802.15.4 y además si están incluidas en el código del MAC del proyecto INETMANET o sólo preparadas “c”, si son opcionales según el estándar IEEE.802.15.7 para un dispositivo con funciones reducidas (RFD) indicado con una “R” u opcionales para ambos tipos RFD y FFD (*full-function devices*) indicado con una “F”, y en algunos casos una breve descripción y notas sobre su uso en el código.

**Tabla 5.1** Lista de primitivas *Ieee802157Mac*. (Sí: X; No: --; completar: c)

Primitiva	Usada	Presente en 802.15.4 / Inetmanet	Opcional RFD / FDD	Nota
<i>mlme_rx_enable_confirm_STEP = 0;</i>	X	-- / --		
<i>mlme_scan_request = false;</i>	X	X / c		
<i>mlme_scan_request_STEP = 0;</i>		-- / c		
<i>mlme_scan_confirm = false;</i>		X / --		
<i>mlme_scan_confirm_STEP = 0;</i>		-- / --		
<i>mlme_start_request = false;</i>		X / c	R	
<i>mlme_start_request_STEP = 0;</i>		-- / c		
<i>mlme_start_confirm = false;</i>		X / --	R	
<i>mlme_start_confirm_STEP = 0;</i>		-- / --		
<i>mlme_sync_request = false;</i>		X / c	F	
<i>mlme_sync_request_STEP = 0;</i>		-- / c		
<i>mlme_sync_request_tracking = false;</i>				En ninguno de los estándares.
<i>mlme_sync_loss_indication = false;</i>		X / --		
<i>mlme_sync_loss_indication_STEP = 0;</i>		-- / --		
<i>mlme_poll_request = false;</i>		X / c		
<i>mlme_poll_request_STEP = 0;</i>		-- / c		
<i>mlme_poll_confirm = false;</i>		X / --		
<i>mlme_poll_confirm_STEP = 0;</i>		-- / --		
<i>CCA_csmaca = false;</i>	X	-- / X		No es primitiva de servicio en el estándar
<i>CCA_csmaca_STEP = 0;</i>	X	-- / X		
<i>RX_ON_csmaca = false;</i>	X	-- / X		No es primitiva de servicio en el estándar
<i>RX_ON_csmaca_STEP = 0;</i>	X	-- / X		

Tabla 5.1 Lista de primitivas *Ieee802157Mac*. (Sí: X; No: --; completar: c) (continuación)

Primitiva	Usada	Presente en 802.15.4 / Inetmanet	Opcional RFD / FDD	Nota
<i>mcps_data_request = false;</i>	X	X / X		
<i>mcps_data_request_STEP = 0;</i>	X	-- / X		
<i>mcps_data_confirm = false;</i>		X / --		
<i>mcps_data_confirm_STEP = 0;</i>		-- / --		
<i>mcps_data_indication = false;</i>		X / --		
<i>mcps_data_indication_STEP = 0;</i>		-- / --		
<i>mcps_purge_request = false;</i>		X / --	F	Opcional para cualquier dispositivo
<i>mcps_purge_request_STEP = 0;</i>		-- / --		
<i>mcps_purge_confirm = false;</i>		X / --	F	Opcional para cualquier dispositivo
<i>mcps_purge_confirm_STEP = 0;</i>		-- / --		
<i>mlme_associate_request = false;</i>		X / c		
<i>mlme_associate_request_STEP = 0;</i>		-- / c		
<i>mlme_associate_indication = false;</i>		X / --	R	
<i>mlme_associate_indication_STEP = 0;</i>		-- / --		
<i>mlme_associate_response = false;</i>		X / c	R	
<i>mlme_associate_response_STEP = 0;</i>		-- / c		
<i>mlme_associate_confirm = false;</i>		X / --		
<i>mlme_associate_confirm_STEP = 0;</i>		-- / --		
<i>mlme_disassociate_request = false;</i>		X / c		
<i>mlme_disassociate_request_STEP = 0;</i>		-- / c		
<i>mlme_disassociate_indication = false;</i>		X / --		
<i>mlme_disassociate_indication_STEP = 0;</i>		-- / --		
<i>mlme_disassociate_confirm = false;</i>		X / --		
<i>mlme_disassociate_confirm_STEP = 0;</i>		-- / --		
<i>mlme_beacon_notify_indication = false;</i>		X / --		
<i>mlme_beacon_notify_indication_STEP = 0;</i>		-- / --		
<i>mlme_get_request = false;</i>		X / --		
<i>mlme_get_request_STEP = 0;</i>		-- / --		
<i>mlme_get_confirm = false;</i>		X / --		
<i>mlme_get_confirm_STEP = 0;</i>		-- / --		
<i>mlme_gts_request = false;</i>		X / --		
<i>mlme_gts_request_STEP = 0;</i>		-- / --		
<i>mlme_gts_indication = false;</i>		X / --		
<i>mlme_gts_indication_STEP = 0;</i>		-- / --		
<i>mlme_gts_confirm = false;</i>		X / --		
<i>mlme_gts_confirm_STEP = 0;</i>		-- / --		
<i>mlme_reset_request = false;</i>		X / c		
<i>mlme_reset_request_STEP = 0;</i>		-- / c		
<i>mlme_reset_confirm = false;</i>		X / --		
<i>mlme_reset_confirm_STEP = 0;</i>		-- / --		
<i>mlme_rx_enable_request = false;</i>		X / c		
<i>mlme_rx_enable_request_STEP = 0;</i>		-- / c		
<i>mlme_rx_enable_confirm = false;</i>		X / --		

En el estándar IEEE.802.15.4, a diferencia de VLC algunas primitivas como las de PURGE son opcionales solo para los dispositivos RFD y no para FFD (full-function devices). Se han eliminado, las comentadas o preparadas en el proyecto para WPAN las primitivas ORPHAN (ver **Tabla 5.1**), pues están fuera del estándar VLC. Otras primitivas no preparadas pero que también difieren de un estándar a otro son COMM-STATUS, DPS, SOUNDING y CALIBRATE.

Dentro de dicha estructura se encuentra la llamada a cada tarea según el tipo enumerado *Ieee802157MacTaskType*, del cual se ha eliminado el miembro TP\_MLME\_ORPHAN\_REQUEST, así como todos los comentarios respectivos a dicha función.

Debido a que desaparecen los tipos de exploración ED y ORPHAN en VLC, es decir, son propios del estándar de sensores IEEE802.15.4, se eliminan también las dos opciones de tipos de exploración en “Ieee802157\_scan”, así como en la implementación de WPAN de INET todas las funciones, mensajes y tramas de comando, referidas al tipo de trama para el comando ORPHAN SCAN. En consecuencia también las primitivas ORPHAN y el tipo de tareas respectivo TP\_MLME\_ORPHAN\_REQUEST.

Para eliminar dicho comando, “Ieee802157\_scan”, se efectúan los siguientes cambios:

1. Eliminación de la constante SIZE\_OF\_802157\_ORPHAN\_NOTIFICATION que define el número de bytes dentro de la cabecera (**Figura 5.2**).
2. Eliminación del tipo de exploración de canal ORPHAN\_SCAN (**Figura 5.3**).
3. Eliminación del tipo comando MAC Ieee802157\_ORPHAN\_NOTIFICATION (**Figura 5.4**).
4. Eliminación del tipo de tarea MAC TP\_MLME\_ORPHAN\_RESPONSE (**Figura 5.5**).
5. Eliminación de la trama de comando Ieee802157OrphanNotiCmd (**Figura 5.6**).
6. Eliminación del mensaje de tramas de comando Ieee802157OrphanNotiCmd (**Figura 5.7**).
7. Modificaciones en la clase que define el nivel MAC, Ieee802157Mac, para eliminar la función ORPHAN:
  - En el método “handleCommand” (**Figura 5.8**).
  - En el método “calFrmByteLength” (**Figura 5.9**).

Local history: Ieee802157Const.h 13-dic-2013 8:25:22	Ieee802157Const.h 70b9de4... (Carolina Martínez)
#define SIZE_OF_802157_ASSOCIATION_RESPONSE 29 // Fig 50: MHR	#define SIZE_OF_802157_ASSOCIATION_RESPONSE 29 // Fig 50:
#define SIZE_OF_802157_DISASSOCIATION_NOTIFICATION 21 // Fig 51: MHR	#define SIZE_OF_802157_DISASSOCIATION_NOTIFICATION 21 // Fig 51:
#define SIZE_OF_802157_PANID_CONFLICT_NOTIFICATION 26 // Fig 53: MHR	#define SIZE_OF_802157_PANID_CONFLICT_NOTIFICATION 26 // Fig 53:
#define SIZE_OF_802157_BEACON_REQUEST 10 // Fig 55: MHR	#define SIZE_OF_802157_ORPHAN_NOTIFICATION 20 // Fig 54:
#define SIZE_OF_802157_GTS_REQUEST 11 // Fig 57: MHR	#define SIZE_OF_802157_BEACON_REQUEST 10 // Fig 55:
#define SIZE_OF_802157_ACK 5 // Fig 46: MHR	#define SIZE_OF_802157_GTS_REQUEST 11 // Fig 57:
	#define SIZE_OF_802157_ACK 5 // Fig 46:

**Figura 5.2.** Eliminación de SIZE\_OF\_802157\_ORPHAN\_NOTIFICATION.

Local history: Ieee802157Enum.h 13-dic-2013 8:25:28	Ieee802157Enum.h 70b9de4... (Carolina Martínez)
// Channel scan type - table 53	// Channel scan type - table 53
typedef enum	typedef enum
{	{
//UNDEFINED = 0,	//UNDEFINED = 0,
ACTIVE_SCAN = 1,	ED_SCAN = 1,
PASSIVE_SCAN = 2	ACTIVE_SCAN = 2,
} Ieee802157ChannelScanType;	PASSIVE_SCAN = 2,
	ORPHAN_SCAN = 4
// MAC task pending type	} Ieee802157ChannelScanType;

**Figura 5.3.** Eliminación del tipo ORPHAN\_SCAN en tipo de exploración.

<pre>Local history: ieee802157Mac.cc 13-dic-2013 8:26:47 case Ieee802157_PANID_CONFLICT_NOTIFICATION: //PAN ID conflict // TBD break;  // case Ieee802157_ORPHAN_NOTIFICATION: //Orphan notification // TBD //wph = HDR_LRvlc(p); sscs-&gt;MLME_ORPHAN_indication(wph-&gt;MHR_SrcAddrInfo.addr 64.fals</pre>	<pre>ieee802157Mac.cc 70b9de4... (Carolina Martínez) case Ieee802157_PANID_CONFLICT_NOTIFICATION: //PAN ID confli // TBD break;  // case Ieee802157_ORPHAN_NOTIFICATION: //Orphan notification // TBD //wph = HDR_LRvlc(p); sscs-&gt;MLME_ORPHAN_indication(wph-&gt;MHR_SrcAddrInfo.addr 64.fals</pre>
<pre>Local history: ieee802157Mac.cc 13-dic-2013 8:26:47 case Ieee802157_PANID_CONFLICT_NOTIFICATION: byteLength = SIZE_OF_802157_PANID_CONFLICT_NOTIFICATION; break;  case Ieee802157_BEACON_REQUEST: byteLength = SIZE_OF_802157_BEACON_REQUEST; break;</pre>	<pre>ieee802157Mac.cc 70b9de4... (Carolina Martínez) case Ieee802157_PANID_CONFLICT_NOTIFICATION: byteLength = SIZE_OF_802157_PANID_CONFLICT_NOTIFICATION; break;  case Ieee802157_ORPHAN_NOTIFICATION: byteLength = SIZE_OF_802157_ORPHAN_NOTIFICATION; break;</pre>

Figura 5.4. Eliminación del tipo de comando Ieee802157\_ORPHAN\_NOTIFICATION.

<pre>Local history: ieee802157Enum.h 13-dic-2013 8:25:28 // MAC command type - Table 67 typedef enum { //Ieee802157_UNDEFINED_CMD = 0, Ieee802157_ASSOCIATION_REQUEST = 1, Ieee802157_ASSOCIATION_RESPONSE = 2, Ieee802157_DISASSOCIATION_NOTIFICATION = 3, Ieee802157_DATA_REQUEST = 4, Ieee802157_PANID_CONFLICT_NOTIFICATION = 5, Ieee802157_BEACON_REQUEST = 6, Ieee802157_COORDINATOR_REALIGNMENT = 7, Ieee802157_GTS_REQUEST = 8 } Ieee802157MacCmdType;</pre>	<pre>ieee802157Enum.h 70b9de4... (Carolina Martínez) // MAC command type - Table 67 typedef enum { //Ieee802157_UNDEFINED_CMD = 0, Ieee802157_ASSOCIATION_REQUEST = 1, Ieee802157_ASSOCIATION_RESPONSE = 2, Ieee802157_DISASSOCIATION_NOTIFICATION = 3, Ieee802157_DATA_REQUEST = 4, Ieee802157_PANID_CONFLICT_NOTIFICATION = 5, Ieee802157_ORPHAN_NOTIFICATION = 6, Ieee802157_BEACON_REQUEST = 7, Ieee802157_COORDINATOR_REALIGNMENT = 8, Ieee802157_GTS_REQUEST = 9 } Ieee802157MacCmdType;</pre>
--	--

Figura 5.5. Eliminación del tipo de tarea TP\_MLME\_ORPHAN\_RESPONSE.

<pre>Local history: ieee802157Frame_m.h 13-dic-2013 8:25:28 // field getter/setter methods };  inline void doPacking(cCommBuffer *b, Ieee802157PanIdConflictNotiCmd&amp; inline void doUnpacking(cCommBuffer *b, Ieee802157PanIdConflictNotiCmd&amp;  /**  * Class generated from &lt;tt&gt;linklayer/ieee802157/contract/Ieee802157Fr  * &lt;pre&gt;  * message Ieee802157BeaconReqCmd extends Ieee802157CmdFrame  * {  * }  * &lt;/pre&gt;  */ class Ieee802157BeaconReqCmd : public ::Ieee802157CmdFrame { protected:</pre>	<pre>ieee802157Frame_m.h 70b9de4... (Carolina Martínez) // field getter/setter methods };  inline void doPacking(cCommBuffer *b, Ieee802157PanIdConflictNotiCmd&amp; inline void doUnpacking(cCommBuffer *b, Ieee802157PanIdConflictNotiCmd&amp;  /**  * Class generated from &lt;tt&gt;linklayer/ieee802157/contract/Ieee802157Fr  * &lt;pre&gt;  * message Ieee802157OrphanNotiCmd extends Ieee802157CmdFrame  * {  * }  * &lt;/pre&gt;  */ class Ieee802157OrphanNotiCmd : public ::Ieee802157CmdFrame { protected:</pre>
<pre>Local history: ieee802157Frame_m.cc 13-dic-2013 8:25:28 switch (field) { default: return NULL; } }  Register_Class(Ieee802157BeaconReqCmd);  Ieee802157BeaconReqCmd::Ieee802157BeaconReqCmd(const char *name, int</pre>	<pre>ieee802157Frame_m.cc 70b9de4... (Carolina Martínez) switch (field) { default: return NULL; } }  Register_Class(Ieee802157OrphanNotiCmd);  Ieee802157OrphanNotiCmd::Ieee802157OrphanNotiCmd(const char *name, int</pre>

Figura 5.6. Eliminación del comando Ieee802157OrphanNotiCmd.

Local history: Ieee802157Frame.msg 13-dic-2013 8:25:28	Ieee802157Frame.msg 70b9de4... (Carolina Martínez)
<pre> } // *** PAN ID conflict notification command - Fig 53 *** message Ieee802157PanIdConflictNotiCmd extends Ieee802157CmdFrame { }  // *** Beacon request command - Fig 55 *** message Ieee802157BeaconReqCmd extends Ieee802157CmdFrame { }  // *** Coordinator realignment command (CRC) - Fig 56 *** message Ieee802157CoorRealignCmd extends Ieee802157CmdFrame </pre>	<pre> } // *** PAN ID conflict notification command - Fig 53 *** message Ieee802157PanIdConflictNotiCmd extends Ieee802157CmdFrame { }  // *** Orphan notification command - Fig 54 *** message Ieee802157OrphanNotiCmd extends Ieee802157CmdFrame { }  // *** Beacon request command - Fig 55 *** message Ieee802157BeaconReqCmd extends Ieee802157CmdFrame </pre>

Figura 5.7. Eliminación del mensaje Ieee802157OrphanNotiCmd.

Local history: Ieee802157Mac.cc 13-dic-2013 8:26:47	Ieee802157Mac.cc 70b9de4... (Carolina Martínez)
<pre> case Ieee802157_PANID_CONFLICT_NOTIFICATION: //PAN ID conflict // TBD break;  // case Ieee802157_ORPHAN_NOTIFICATION: //Orphan notification // TBD /*wph = HDR_LRvlc(p); sscs-&gt;MLME_ORPHAN_indication(wph-&gt;MHR_SrcAddrInfo.addr_64,fals // break; </pre>	<pre> case Ieee802157_PANID_CONFLICT_NOTIFICATION: //PAN ID confli // TBD break;  case Ieee802157_ORPHAN_NOTIFICATION: //Orphan notification // TBD /*wph = HDR_LRvlc(p); sscs-&gt;MLME_ORPHAN_indication(wph-&gt;MHR_SrcAddrInfo.addr_64, break; </pre>

Figura 5.8. Método “handleCommand”.

Local history: Ieee802157Mac.cc 13-dic-2013 8:26:47	Ieee802157Mac.cc 70b9de4... (Carolina Martínez)
<pre> case Ieee802157_PANID_CONFLICT_NOTIFICATION: byteLength = SIZE_OF_802157_PANID_CONFLICT_NOTIFICATION; break;  case Ieee802157_BEACON_REQUEST: byteLength = SIZE_OF_802157_BEACON_REQUEST; break; </pre>	<pre> case Ieee802157_PANID_CONFLICT_NOTIFICATION: byteLength = SIZE_OF_802157_PANID_CONFLICT_NOTIFICATION; break;  case Ieee802157_ORPHAN_NOTIFICATION: byteLength = SIZE_OF_802157_ORPHAN_NOTIFICATION; break; </pre>

Figura 5.9. Método “calFrmByteLength”.

### 5.5.2. Formato trama. Cabecera MAC, MHR.

Desaparecen los sub-campos *intraPan* y *battLifeExt*, ver **Figura 5.10**, propios del estándar IEEE.802.15.4 y no presente en IEEE.802.15.7, del campo de control y especificaciones de super-trama respectivamente. Esto implica supresión de los sub-campos tipo de trama (*frmType*) en la clase *Ieee802157Field*, que contiene los campos que almacenan el estado de la estructura del nivel de enlace, y modificaciones en varios de los métodos de la clase MAC.

Se requirió también realizar las siguientes modificaciones en la clase *Ieee802157Mac* que define el nivel MAC para eliminar el campo *intraPan* en tipo de trama:

- En el método “constructACK” (**Figura 5.11**).
- En el método “MCPS\_DATA\_request” (**Figura 5.12**).
- En el método “handleBcnTxTimer” (**Figura 5.13**).

C Compare Viewer	Ieee802157Field.h 13-dic-2013 8:25:21	Ieee802157Field.h 70b9de4... (Carolina Martínez)
	<pre> Ieee802157FrameType frmType; bool secu; bool frmPending; bool ackReq; UINT_8 dstAddrMode; UINT_8 srcAddrMode; }; </pre>	<pre> Ieee802157FrameType frmType; bool secu; bool frmPending; bool ackReq; bool intraPan; UINT_8 dstAddrMode; UINT_8 srcAddrMode; </pre>

Figura 5.10. Eliminación del campo *intraPan* en tipo de trama.

Local history: Ieee802157Mac.cc 13-dic-2013 8:26:47	Ieee802157Mac.cc 70b9de4... (Carolina Martínez)
ackFrmCtrl.frmType = Ieee802157_ACK;	ackFrmCtrl.frmType = Ieee802157_ACK;
ackFrmCtrl.secu = false;	ackFrmCtrl.secu = false;
ackFrmCtrl.frmPending = (i==0)?true:false;	ackFrmCtrl.frmPending = (i==0)?true:false;
ackFrmCtrl.ackReq = false;	ackFrmCtrl.ackReq = false;
ackFrmCtrl.dstAddrMode = defFrmCtrl_AddrModeNone; // dst add	ackFrmCtrl.intraPan = origFrmCtrl.intraPan; //
ackFrmCtrl.srcAddrMode = defFrmCtrl_AddrModeNone; // src add	ackFrmCtrl.dstAddrMode = defFrmCtrl_AddrModeNone; // dst
	ackFrmCtrl.srcAddrMode = defFrmCtrl_AddrModeNone; // src

Figura 5.11. Método “constructACK”.

Local history: Ieee802157Mac.cc 13-dic-2013 8:26:47	Ieee802157Mac.cc 70b9de4... (Carolina Martínez)
// construct frame control field	// construct frame control field
frmCtrl.frmType = Ieee802157_DATA; //data type	frmCtrl.frmType = Ieee802157_DATA; //data type
frmCtrl.secu = secuData;	frmCtrl.secu = secuData;
frmCtrl.frmPending = false;	frmCtrl.frmPending = false;
frmCtrl.dstAddrMode = DstAddrMode;	frmCtrl.intraPan = (SrcPANID == DstPANID)? true:false;
frmCtrl.srcAddrMode = SrcAddrMode;	frmCtrl.dstAddrMode = DstAddrMode;

Figura 5.12. Método “MCPS\_DATA\_request”.

Local history: Ieee802157Mac.cc 13-dic-2013 8:26:47	Ieee802157Mac.cc 70b9de4... (Carolina Martínez)
Ieee802157BeaconFrame* tmpBcn = new Ieee802157BeaconFrameC tmpBcn->setName("Ieee802157BEACON");	Ieee802157BeaconFrame* tmpBcn = new Ieee802157BeaconFr tmpBcn->setName("Ieee802157BEACON");
// construct frame control field	// construct frame control field
frmCtrl.frmType = Ieee802157_BEACON;	frmCtrl.frmType = Ieee802157_BEACON;
frmCtrl.secu = secuBeacon;	frmCtrl.secu = secuBeacon;
frmCtrl.frmPending = false; // TBD: indirect trans	frmCtrl.frmPending = false; // TBD: indirect trans
frmCtrl.ackReq = false; // ignored upon reception	frmCtrl.ackReq = false; // ignored upon recept
frmCtrl.dstAddrMode = defFrmCtrl_AddrModeNone; // 0x00, ig	frmCtrl.intraPan = false; // ignored upon recept
if (mpib.macShortAddress == 0xfffe)	frmCtrl.dstAddrMode = defFrmCtrl_AddrModeNone; // 0x00

Figura 5.13. Método “handleBcnTxTimer”.

De forma metódica y haciendo uso de las facilidades proporcionadas por la herramienta de desarrollo se sustituyen todos los nombres referidos a un estándar por la nomenclatura de VLC, tanto en la clase que define en el nivel MAC, como en el resto de niveles afectados, ficheros de simulación y de configuración de escenarios.

La última fase del acondicionamiento al nuevo estándar fue la inclusión de aquellos elementos, tanto campos y tramas como sus respectivos paquetes de mensajes y tareas del nivel de acceso al medio, que se consideraron necesaria para la implantación de los mecanismos a desarrollar posteriormente.

A continuación se describen las variables introducidas para la implementación de dichos campos y tramas.

### 5.5.3. Todas las tramas

Campos de cabecera comunes añadidos en todas las tramas

- **AuxSecHdr: Auxiliary Security Header**, el campo cabecera de seguridad auxiliar, que tiene una longitud variable, especifica la información requerida para el procedimiento de seguridad, incluyendo como está protegida la trama actual (nivel de seguridad) y qué material de claves de los atributos PIB de MAC están usándose. Este campo aparece si el subcampo de seguridad habilitada del campo de control está al valor '1'.
- **FCS: Frame Check Sequence**, el campo secuencia de verificación de trama es de dos octetos de longitud y se calcula según el tamaño del MHR y carga útil de cada trama con carga mayor a '0'bytes. No está implementado.

### 5.5.4. Tramas de baliza

Campos de las tramas de baliza añadidos:

- **GTSFields:** *GTS fields*, los campos GTS contienen subcampos de información de especificación, dirección y lista de GTSs. El proceso GTS implementado no utiliza la información GTS en las balizas.
- **pendAddrFields:** *Pending Address fields*, los campos de dirección pendiente contienen subcampos de listas de direcciones pendientes, en concreto subcampos de especificaciones y un subcampo de lista de direcciones. Necesario para la transmisión indirecta.
- **cellSrchLength:** *Cell Search Length*, el campo, si está presente (*cellSrchEn='1'*), especifica el número de slots necesarios en el CFP para el tamaño y la búsqueda de ubicación de las celdas. Propio de VLC, para el proceso de búsqueda de ubicación y tamaño de celdas.
- **bcnPayload:** *Beacon Payload*, el campo carga útil de baliza es una secuencia opcional de hasta valor *aMaxBeaconPayloadLength* octetos específicos para ser transmitidos en la trama de baliza por la siguiente capa superior. No está implementado.

### 5.5.5. Tramas de datos

Campos de las tramas de datos añadidos:

- **dataType:** *Data Type*, el campo tipo de dato es de un byte que se define según la **Tabla 5.2**. Menciona la forma utilizada para enviar el dato, individual, paquete o ráfaga, así como el número de PPDU asociadas a esta trama de datos. Este byte se sitúa al principio de la carga. No estaba contemplado en el código original, pues la carga iba directamente al *FrmPayload*.

**Tabla 5.2** Formato del campo de carga útil de trama de datos.

Bits: 0 – 1	Bits: 2 – 7	Variable
00 – Single (Único)	Número de PPDU	<i>Data Payload</i>
01 – <i>Packet</i> (Paquete)	por Trama de datos	(Carga de datos)
10 – <i>Burst</i> (Ráfaga)		
11 – <i>Reserved</i> (Reservado)		

### 5.5.6. Control de trama y supertrama

Subcampos de control de trama y super-trama:

- **frmVersion:** *Frame version*, subcampo que indica el número de versión a la que corresponde la trama. El valor *0b00* indica que es una trama compatible con *IEEE Std 802.15.7*.
- **cellSrchEn:** *Cell Search Enable*, subcampo que indica cuando su valor es '1' que se transmite la longitud.
- **cellSearchLength** como un campo adicional en la trama de baliza.

### 5.5.7. Campos de información GTS

Subcampos dentro del campo de información de GTS:

- **GTSDesCount:** *GTS Descriptor Count*, el subcampo contador de descriptores de GTS especifica el número de descriptores GTS de tres octetos contenidos en el campo lista de GTS de la trama de baliza.
- **GTSPermit:** *GTS Permit*, el subcampo permiso de GTS estará a '1' o a '0' según el valor verdadero o falso del atributo PIB macGTSPermit respectivamente.
- **GTSDirMask:** *GTS Directions Mask*, el subcampo máscara de las orientaciones de GTS contiene una máscara que identifica la orientación, GTS de recepción o GTS de transmisión, de cada GTS de la super-trama. Cada bit corresponde a un GTS y '0' indica que es de transmisión y '1' que es de recepción.
- **GTSListk:** *GTS List*, campo lista de descriptores de GTS contiene tantos descriptores como indica el subcampo de especificaciones contador de descriptores de GTS y de él depende por tanto la longitud de este campo. Cada Descriptor contendrá los siguientes subcampos:
  - **devShortAddr:** *Device Short Address*, el subcampo dirección corta del dispositivo contendrá dicha dirección para la cual el descriptor de GTS está dirigido. Actualmente es el campo del tipo *GTSDescriptor* llamado con el mismo nombre, y se le da valor en el comando de solicitud de GTS, y se utiliza en los dispositivos para saber si el GTS le corresponde.
  - **GTSSlot:** *GTS Starting Slot*, el subcampo slot de comienzo de GTS contiene el slot de super-trama en el cual va a comenzar el GTS al que corresponde el descriptor. Inicialmente era el campo del tipo *GTSDescriptor* llamado *startSlot*, y se le daba valor en el comando de solicitud de GTS, y se utilizaba en la planificación de GTS de la super-trama.
  - **GTSLength:** *GTS Length*, el subcampo longitud de GTS contiene la longitud del GTS al que corresponde el descriptor. Inicialmente era el campo del tipo *GTSDescriptor* llamado *length*, se usaba para el cálculo del slot de comienzo y se le pasaba el valor en el comando de solicitud de GTS.

### 5.5.8. Trama CVD

Se incluye las tramas CVD propias del estándar de comunicación por luz visible, y dentro del subcampo tipo de trama, se añade el tipo de trama correspondiente *Ieee802157\_CVD*.

### 5.5.9. Comando de información

Existen dos comandos propios del estándar VLC que son elementos de información, IE, uno de ellos contiene las prestaciones. Se definieron para éste último los tipos necesarios para el campo de información de prestaciones *Capability Information Field*, referentes a todas las prestaciones de la capa de enlace, la capa física, y la banda utilizada

## 5.6. Modo con baliza habilitada

Durante el análisis de resultados de las simulaciones de comprobación, y atendiendo a los detalles referidos a la capa de acceso al medio, se detecta que para el modo de transmisión con baliza habilitada la configuración de supertrama con el valor del orden de supertrama igual al valor de orden de baliza provoca un error.

La variable en la que se produce el error es el siguiente:

```
/** @brief buffer for beacon frames to be transmitted without CSMA/CA */
ieee802157Frame* txBeacon;
```

El punto del código en el que tiene lugar en el método *handleTxBeaconTimer*, de la capa MAC, cuando es tiempo para transmitir la siguiente baliza pero en el buffer no se ha vaciado correctamente: `ASSERT(txBeacon == NULL);`

Para el caso  $SO=BO$ , ocurre que coinciden los eventos correspondientes a final de supertrama, y transmisión de la siguiente baliza.

Estos eventos *txSDTimer* y *bcnTxTimer*, son programados por los temporizadores, y representan el instante de fin de modo transmisión de la duración de supertrama e instante de transmisión de la siguiente baliza respectivamente. Ambos son incompatibles pues al finalizar el tiempo activo de supertrama, se le indica a la capa física que apague el transceptor para entrar en modo *sleep*, y a la vez, es el momento de transmitir la siguiente baliza, lo que la capa física descarta por estar en dicho modo de suspensión.

El comportamiento en el nodo tras la primera vez que tiene lugar el error en el intento de transmisión de baliza, es que al no recibir la confirmación de tarea de transmisión de baliza con éxito, no se borra la trama de baliza del búfer, por tanto no se envía, y además en el siguiente intento provoca error debido a que el búfer de tramas de baliza no está vacío. Esto causa la pérdida de sincronización del modo de transmisión directo con baliza habilitada, inutilizando todos los periodos activos de la supertrama.

### 5.6.1. Solución #1.

La primera solución consistió en examinar el caso  $BO=SO$  dentro del proceso de gestión del temporizador de duración de super-trama (*handleSDTimer*), es decir, si los parámetros *SO* y *BO* coinciden, no se tendría un modo inactivo, con lo que no se enviaría la primitiva: `PLME_SET_TRX_STATE_request(phy_TRX_OFF)`.

Al incluir esta condición en el código, ya podemos simular el parámetro *SO* con valores menor o igual a *BO*.

Esta solución, mantiene la condición original que incluye que el parámetro *txAck* sea nulo, es decir, que no haya un ACK pendiente de transmitir para apagar el transceptor. En el caso de que lo haya aumenta el parámetro estadístico *numTxAckInactive* y no envía la primitiva de solicitud `PLME_SET_TRX_STATE_request(phy_TRX_OFF)` a la capa física, aunque haya llegado al final de la supertrama.

La consideración de esta solución, es que si es el final de supertrama, se apague el transceptor con independencia de que *txACK* sea distinto de NULL, a excepción de que *SO* y *BO* sean iguales. Por tanto, se crea una segunda solución, descrita a continuación.

### 5.6.2. Solución #2.

Esta segunda solución, deja la posibilidad de que llegue al final de supertrama con *txACK* distinto de NULL controlada con un mensaje de error, e independiza la condición del modo *sleep* de ello.

### 5.6.3. Pérdida de sincronización.

Otro de los problemas en los que se ha trabajado es la pérdida de sincronización respecto a las tramas beacon transmitidas por el coordinador. Esto parece ser debido, por un lado, a una diferencia temporal producida por los cálculos para el tiempo de *turnaround* del transceptor, al pasar de modo recepción a modo transmisión, que se requieren en el coordinador y no en los dispositivos. Esta disparidad a la hora de realizar dichos cálculos temporales, provoca una diferencia entre la recepción de baliza y la espera de dicha recepción en los dispositivos, debido al redondeo y truncamiento. Además, el tiempo de *turnaround* no está siendo considerado en el cálculo temporal de duración de la parte activa de la supertrama en los dispositivos, lo que puede suponer pérdidas de tramas transmitidas durante un tiempo fuera del tiempo real de supertrama activa establecido por el coordinador para operar, entre otros problemas.

Para aclarar los distintos errores de cálculo ocasionados debidos a la misma diferencia temporal en varias variables y bajo distintas condiciones, enumeraremos los casos de la siguiente forma:

- **Caso 1:** El tiempo de recepción de una trama beacon en los dispositivos no corresponde con la recepción de ésta. Ver apartado 5.6.3.1.
- **Caso 2:** El tiempo de fin de supertrama activa es distinto en el coordinador y en los dispositivos. Ver apartado 5.6.3.2.
- **Caso 3:** Cuando *BI* y *SD* deberían ser iguales, hay errores de truncamiento que crean diferencias entre estos valores.

Para el primer caso, debemos entender qué debería suceder y cuáles son los métodos y cálculos implicados.

Las variables globales con las que se planifica tanto la transmisión de la siguiente baliza como la recepción de ésta son *bcnTxTimer* y *bcnRxTimer* respectivamente, calculadas en sus correspondientes métodos de inicio de temporizadores *startBcnTxTimer* y *startBcnRxTimer*. En el caso de topología estrella, el coordinador se encargará sólo de transmitir beacons y los dispositivos sólo de recibirlos, por lo que cada uno usará el método adecuado.

#### 5.6.3.1. Caso #1.

1. **Método para iniciar el temporizador de transmisión de un beacon: startBcnTxTimer**

**bcnTxTime:** variable global para el instante de tiempo de transmisión de baliza, en el cual o bien debe solicitar el modo transmisión al transceptor a la capa PHY y recalcularse, o bien ya es el instante de comienzo de transmisión de la trama.

En el caso de no transmisión: Calcula el tiempo de inicio de transmisión de la baliza y le resta el tiempo de cambio de modo del transceptor del peor caso, de recepción a transmisión. Esto lo hace una primera vez, según la siguiente expresión (5.1):

$$w\_time = \frac{(aBaseSuperframeDuration \cdot 2^{macBeaconOrder}) - aTurnaroundTimeRxTx}{phy\_mode\_min\_clk} \quad (5.1)$$

donde *aBaseSuperframeDuration* es la duración de la supertrama que se obtiene como *aBaseSlotDuration\_aNumSuperframeSlots*(60·16 slots) relojes ópticos,

*macBeaconOrder* el atributo *PIB* recogido de las características de red iniciales en el caso del coordinador como parámetro *BO*, «beacon order»; *phy\_mode\_min\_clk* es el mínimo reloj usado en el dispositivo (relojes ópticos/s).

La variable *w\_time* miembro temporal que recoge el tiempo en el que se programa esta primera vez **bcnTxTimer**, que en este caso será el tiempo total de final del intervalo entre balizas adelantado un tiempo igual a (*aTurnaroundTimeRxTx/phy\_mode\_min\_clk*) segundos.

Cuando el temporizador planificado con el tiempo anterior se cumple, planifica ahora en el instante: *now + (aTurnaroundTimeRxTx/phy\_mode\_min\_clk)*.

Este es el tiempo de final de intervalo entre balizas o *BI*, y el instante de transmisión de la siguiente baliza, y debería ser igual a (*aBaseSuperframeDuration·2<sup>BO</sup>/phy\_mode\_min\_clk*).

Por otro lado, el cálculo cuando es la primera transmisión de trama beacon, si *BO* es distinto de 15, es decir el dispositivo es con baliza habilitada, entonces transmite inmediatamente, y si *BO*=15 se realizan otros cálculos que no afectan al caso que estamos estudiando.

## 2. Método para iniciar el temporizador de recepción de un beacon: **startBcnRxTimer**

**bcnRxTime**: variable global para el instante de tiempo de recepción de baliza, en el cual el dispositivo debe solicitar el modo receptor en el transceptor a la capa PHY. Primero calcula la duración del intervalo entre beacons según la expresión (5.2):

$$rxBI = \frac{BI}{phy\_mode\_min\_clk} \quad (5.2)$$

siendo *BI* un parámetro recogido de la trama beacon y que corresponde al siguiente cálculo (5.3) en el coordinador:

$$BI = aBaseSuperframeDuration + 2^{macBeaconOrder} \quad (5.3)$$

A continuación calcula el tiempo transcurrido desde que debió comenzar la recepción de la última baliza, con *t\_bcnRxTime* calculado a partir de la variable *bcnRxTime*, actualizada con la recepción de baliza y teniendo en cuenta la posible pérdida de recepción de alguna baliza, *tmpf*, según (5.4):

$$tmpf = now - t\_bcnRxTime \quad (5.4)$$

siendo *now* el tiempo de simulación actual.

Luego calcula el tiempo restante hasta la próxima recepción de beacon (5.5):

$$w\_time = rxBI - tmpf \quad (5.5)$$

Y el tiempo absoluto hasta el instante de recepción, *tmpf*, según (5.6):

$$tmpf = now + w\_time \quad (5.6)$$

Si el tiempo en el que se espera la siguiente baliza menos el tiempo en el que se programó la espera de recepción de la anterior es menor que la duración del intervalo de baliza (*BI*), entonces hay pérdida de sincronismo y en ese caso, vuelve a calcular *w\_time* de la siguiente forma (5.7):

$$w\_time = 2 \cdot rxBI + t\_bcnRxTime - now \quad (5.7)$$

Este tiempo sería el tiempo absoluto para la recepción de dos balizas en adelante, es decir, el fallo supondría la pérdida de la próxima baliza, y se reajustarían los tiempos de recepción para dos más adelante.

Finalmente, el método **startBcnRxTimer** finalmente planificará la recepción de baliza para el dispositivo según el valor final de *w\_time*, que será, o el calculado inicialmente o en el caso excepcional anteriormente comentado, en la variable global **bcnRxTimer**.

### 3. Descripción del problema

El temporizador que recoge el tiempo para recibir una baliza, no realiza el cálculo del mismo modo que en el coordinador, un error de truncamiento provoca una diferencia temporal mayor que el retardo debido a la transmisión en el medio, es decir, el valor de **bcnRxTimer** si debe ser mayor al valor **bcnTxTimer**, pero con una diferencia algo menor (diferencia de 1ps para PHYII y 3,75MHz).

El código implementado tiene una condición para el caso en que si la diferencia entre el valor temporal planificado para la siguiente recepción y el tiempo en el que se planificó la recepción anterior (*lastTime\_bcnRxTimer*) es menor al valor de duración de intervalo entre balizas (*rxBI*), se haga un nuevo cálculo a partir de *rxBI* y el tiempo en que realmente recibió la baliza (**bcnRxTime**), esto puede ocurrir si el dispositivo casualmente ya estaba en modo recepción en el momento en que el coordinador transmitió el beacon, por lo que es capaz de percibir el error ya que los valores **bcnRxTime** (tiempo en que se recibió la baliza) y *lastTime\_bcnRxTimer* (tiempo en el que se planificó recibirla) difieren. Este nuevo cálculo bajo dicha condición, lo hace para dos intervalos entre balizas, es decir, para corregir la pérdida de sincronización, se pierde la recepción de una baliza. Además, el error volvería a darse.

Los métodos asociados al proceso CSMA/CA, realizan los cálculos con el valor de recepción de baliza, por lo que no es problema el error de planificación del evento de recepción de siguiente beacon.

#### 5.6.3.2. Caso #2.

1. Método para iniciar el temporizador de fin de supertrama del coordinador: **startTxSDTimer**

**txSDTimer**: variable global para el instante de tiempo de fin de supertrama, en el cual el dispositivo coordinador debe solicitar apagar el transceptor a la capa PHY, exceptuando el caso en el que *BO* y *SO* sean idénticos y sea el momento de transmitir un beacon.

Calcula el tiempo de duración de supertrama de la siguiente forma (5.8):

$$wtime = \frac{aNumSuperframeSlots \cdot txSfSlotDuration}{phy\_mode\_min\_clk} \quad (5.8)$$

siendo *txSfSlotDuration* un parámetro calculado al iniciar el coordinador y que corresponde al siguiente cálculo (5.9):

$$txSfSlotDuration = aBaseSlotDuration + 2^{macSuperframeOrder} \quad (5.9)$$

El tiempo absoluto calculado para el final de supertrama será programado en la variable global *txSDTimer* con el valor *now+wtime*, siendo *now* el tiempo de simulación actual.

## 2. Método para iniciar el temporizador de fin de supertrama del dispositivo: startRxSDTimer

**rxSDTimer:** variable global para el instante de tiempo de fin de supertrama, en el cual el dispositivo debe solicitar apagar el transceptor a la capa PHY.

Primero calcula el tiempo de duración de supertrama de la siguiente forma (5.10):

$$wtime = \frac{aNumSuperframeSlots \cdot rxSfSlotDuration}{phy\_mode\_min\_clk} \quad (5.10)$$

siendo *rxSfSlotDuration* un parámetro calculado al llegar una trama beacon y que corresponde al siguiente cálculo (5.11):

$$rxSfSlotDuration = aBaseSlotDuration + 2^{rxSO} \quad (5.11)$$

Luego calcula el tiempo que queda para el final de la supertrama, añadiendo el tiempo en que llegó el último beacon y restando el tiempo actual de simulación:

$$wtime = 2 \cdot rxBI + bcnRxTime - now \quad (5.12)$$

Tras asegurarse que el cálculo anterior es mayor que cero, planifica el evento para el final de supertrama en la variable global *rxSDTimer* con el valor *now+wtime*.

## 3. Descripción del problema

No debería surgir problema en la transmisión hasta el fin de la supertrama activa para el dispositivo, pues la diferencia de 3333ps debido al retardo de transmisión en el medio, es decir, por el valor *beaconRxTime* el cual se debe tener en cuenta para la posible recepción de tramas transmitidas por el coordinador, aunque no se tiene en cuenta en la dirección opuesta, supone menos de un periodo de backoff en cualquier caso de *phy\_mode\_min\_clk*.

## 5.7. Mecanismo de acceso múltiple por detección de portadora con y sin evasión de colisiones

Con el fin de implementar el mecanismo de acceso al medio fue necesario llevar a cabo dos grandes desarrollos: una máquina de estados finitos (FSM) y un algoritmo de acceso al canal. Se describen ambos desarrollos a continuación.

### 5.7.1. FSM

En primer lugar se hizo preciso implementar un modelo de estados finitos o *Finite State Model* (FSM) para la transmisión de trama de datos con CSMA-CA. El proceso de transmisión de una trama de datos se controla por medio de una máquina de estados finitos, cuyos estados son de espera, inicialización del proceso, control de acceso libre al canal y transmisión, y control de recepción de ACK.

Los elementos básicos utilizados en una FSM para describir un sistema son: entradas, salidas, funciones de transición entre estados y funciones para generación de los valores de salida. Este modelo se describe en el **Anexo II**.

## 5.7.2. Algoritmo de acceso al canal

El algoritmo de acceso al canal desarrollado en este PFC se basa en el algoritmo incluido por INETMANET para la norma IEEE802.15.4. Fue necesario estudiar las diferencias con respecto al modelo necesario para VLC, detallando las deficiencias a cubrir, y los cambios a realizar. Este análisis se describe en detalle en el **Anexo III**.

A continuación se presentan las sucesivas propuestas realizadas en este PFC para desarrollar el modelo de acceso al canal, y los cambios que cada propuesta representa.

### 5.7.2.1. Propuesta 01

Se propone mantener el código según la versión de INETMANET, que incluye la ventana de contención y carece de la opción de deshabilitar el CCA, modificando el método “Ieee802157Mac::FSM\_MCPS\_DATA\_request” para incluir el aumento de *NB* (número de backoff) en el caso de no recibir el ACK requerido y volver al proceso de reenvío de la trama.

#### 5.7.2.1.1. Cambio 01.

Cuando termina el tiempo de espera de acuse de recibo correspondiente a la trama enviada con éxito previamente, la máquina de estados finitos se encarga de comprobar que el número de intentos de reenvíos de trama no es el máximo establecido, esta versión incluiría comprobar a continuación que tampoco supere el valor máximo de periodos de backoff. Pero cada vez que vuelve al método “csmacaResume” se resetea el valor NB con la llamada a su vez al método “csmacaReset”.

Por tanto esta propuesta únicamente supondría un cambio en el caso de haber existido en el primer intento de transmisión de datos 4 intentos de tiempos de backoff y al no recibir ACK, el valor máximo macMaxCSMABackoffs no permitiera un reintento del proceso. La **Figura 5.14** ilustra el cambio 01 de la propuesta 01.

```

C Compare Viewer Differences shown might not be optimal
Local history: Ieee802157Mac.cc 16-ene-2014 8:42:23
Ieee802157Mac.cc 35ef3fa... (Carolina Martínez)

strncpy(taskP.taskFrFunc(task), "csmacaCallBack");
waitDataAck = false;
//CAMBIOS IEEE802157: Test NB variation when ACK f
//The random channel access in IEEE802157 must cor
//This involves changing the process of Resume ad
//csmacaResume();
NB++;
if (NB > mpib.macMaxCSMABackoffs) // Case F1
{
    tmpCsmaca = 0;
    csmacaCallBack(phy_BUSY);
    return;
}
else // Case F2: backoff again
{
    BE++;
    if (BE > aMaxBE) BE = aMaxBE; // aMaxBE
    csmacaResume();
    //csmacaStart(false); // not the f
}
//FIN CAMBIOS IEEE802157

strncpy(taskP.taskFrFunc(task), "csmacaCallBack");
waitDataAck = false;
csmacaResume();
}
else
{
    EV << "[MAC]: the max num of retries reached"
    taskP.taskStatus(task) = false; //task fails
    EV << "[MAC-TASK-FAIL]: reset TP_MCPS_DATA_REC
    waitDataAck = false; // debug
    resetTRX();
    taskFailed('d', mac_NO_ACK);
}
}
break;
default:
break;
}
}
else if (txOption == INDIRECT_TRANS)
{

```

**Figura 5.14.** Cambio 01 de la propuesta 01: contabilizar los reintentos de Backoff también dentro de la FSM.

### 5.7.2.2. Propuesta 02

Se propone mantener el cambio de la propuesta anterior, y cambiar la llamada al método “*csmacaReset*” de inicialización de los parámetros *NB*, *CW* y *BE*, pues según la norma este método sólo se llamaría al inicio del proceso.

#### 5.7.2.2.1. Cambio 01.

Se propone llamar al “*csmacaReset*” en el método de entrada de trama de transmisión “*csmacaEntry*”, en lugar de en “*csmacaStart*” cuando la condición de “*firsttime*” se cumple. Este cambio se muestra en la **Figura 5.15**.

```

Local: ieee802157Mac.cc
else //if (pktType == 'd') //txData
{
    waitDataAck = false; //data packet not yet transmit
    numDataRetry = 0;
    //CAMBIOS IEEE802157: Random channel access VLC version.
    csmacaReset(beaconEnabled);
    //FIN CAMBIOS IEEE802157
    csmacaResume();
}

ieee802157Mac.cc f1b66c7... (Carolina Martínez)
else //if (pktType == 'd') //txData
{
    waitDataAck = false; //data packet not yet tran
    numDataRetry = 0;
    csmacaResume();
}

void Ieee802157Mac::csmacaResume()
{
    bool ackReq;
}
    
```

Figura 5.15. Cambio 01 de la propuesta 02: llamada a “*csmacaReset*” en el método “*csmacaEntry*”.

#### 5.7.2.2.2. Cambio 02.

Comentar la llamada a “*csmacaReset*” en el método “*csmacaStart*” cuando la condición de “*firsttime*” se cumple. De esta forma, en cualquier reintento de transmisión de la trama de datos no se iniciarían los valores *NB*, *CW*, y *BE*, manteniendo por tanto el valor obtenido durante el proceso CSMA-CA de la misma. Por tanto el valor máximo de tiempos. Este cambio se muestra en la **Figura 5.16**.

```

Local: ieee802157Mac.cc
if (firsttime)
{
    EV << "[CSMA-CA]: starting CSMA-CA for " << frame->getName() << endl;
    beaconEnabled = ((mpib.macBeaconOrder != 15) || (rxB0 != 15));
    //CAMBIOS IEEE802157: Random channel access VLC version.
    //csmacaReset(beaconEnabled);
    //FIN CAMBIOS IEEE802157
    ASSERT(tmpCsmaca == 0);
    tmpCsmaca = frame;
    csmacaAckReq = ackReq;
}

ieee802157Mac.cc f1b66c7... (Carolina Martínez)
if (firsttime)
{
    EV << "[CSMA-CA]: starting CSMA-CA for " << frame->getName() << endl;
    beaconEnabled = ((mpib.macBeaconOrder != 15) || (rxB0 != 15));
    //csmacaReset(beaconEnabled);
    ASSERT(tmpCsmaca == 0);
    tmpCsmaca = frame;
    csmacaAckReq = ackReq;
}

if (rxB0 != 15) // is receiving beacon
    
```

Figura 5.16. Cambio 02 de la propuesta 02: eliminar la llamada a “*csmacaReset*” en el método “*csmacaStart*”.

### 5.7.2.3. Propuesta 03

Esta propuesta culmina la adaptación del algoritmo. Se precisaron cinco cambios.

#### 5.7.2.3.1. Cambio 01.

La inicialización es lo primero que se realiza, la diferencia entre ambos protocolos es que es igual tanto para acceso ranurado como no ranurado, eso hace que no sea

necesario el parámetro de entrada `beaconEnable` para el método “`Ieee802157Mac::csmacaReset`” encargado únicamente de inicializar los valores.

Este cambio se muestra en el **Código 5.1** siguiente.

**Código 5.1.** Código inicial del método `csmacaReset`.

```

/*
 * Ieee802157Mac::csmacaReset
 * Actual
 */

void Ieee802157Mac::csmacaReset(bool bcnEnabled)
{
    if (bcnEnabled)
    {
        NB = 0;
        CW = 2;
        BE = mpib.macMinBE;
        if ((mpib.macBattLifeExt)&&(BE > 2))
            BE = 2;
    }
    else
    {
        NB = 0;
        BE = mpib.macMinBE;
    }
}
/*
Cambio para VLC:
/* * CAMBIOS IEEE802157: New csmacaReset of Random Access
Algorithm.
 * Both CW as macBattLifeExt consideration disappear.
 * The beaconEnabled argument is no longer necessary.
 * Same initialization for slotted or unslotted access.
**/
*/

{
    NB = 0;
    BE = mpib.macMinBE;
}

```

### 5.7.2.3.2. Cambio 02.

Como consecuencia del cambio01 en el comienzo del algoritmo del cual se encarga el método “`Ieee802157Mac::csmacaStart`” únicamente cambiamos la llamada al método “`Ieee802157Mac::csmacaReset`” con las nuevas características, es decir sin argumento de entrada. Este cambio se muestra en el **Código 5.2**

**Código 5.2.** Código para *Propuesta 03. Cambio 02.*

```
/*  
 * Ieee802157Mac::csmacaStart  
 * Actual  
 */  
  
csmacaReset(beaconEnabled);  
  
/*  
 Cambio para VLC:  
 /* * CAMBIOS IEEE802157: New csmacaReset of Random Access  
 Algorithm.  
 * Both CW as macBattLifeExt consideration disappear.  
 * The beaconEnabled argument is no longer necessary.  
 * Same initialization for slotted or unslotted access.  
 **/  
 */  
  
csmacaReset();
```

---

### 5.7.2.3.3. Cambio 03.

El cambio más significativo está después de recibir la respuesta de la capa PHY respecto al estado del canal. En el caso de VLC, el proceso se simplifica al carecer de ventana de contención. La capa MAC envía una primitiva al nivel físico solicitando la escucha del canal y recibe la confirmación con el estado *bussy* o *iddle* según esté ocupado o libre en una primitiva en sentido opuesto, ésta se gestiona en el método “Ieee802157Mac::handle\_PLME\_CCA\_confirm”, en el cual realizaremos los cambios para VLC.

Este cambio se muestra en el **Código 5.3** y **5.4**.

Código 5.3. Código para *Propuesta 03. Cambio 03.*

```

/*
 * Ieee802157Mac::handle_PLME_CCA_confirm
 * Actual
 */

void Ieee802157Mac::handle_PLME_CCA_confirm(PHYenum status)
{
    if (taskP.taskStatus(TP_CCA_CSMACA))
    {
        taskP.taskStatus(TP_CCA_CSMACA) = false;
        // the following from Csmaca802_15_4::CCA_confirm
        if (status == phy_IDLE) // idle channel
        {
            if (!beaconEnabled) // non-beacon, unslotted
            {
                tmpCsmaca = NULL;
                csmacaCallback(phy_IDLE);
                // unslotted successfully, callback
                return;
            }
            else // beacon-enabled, slotted
            {
                CW--; // Case C1, beacon-enabled, slotted
                if (CW == 0) // Case D1
                {
                    //timing condition may not still hold - check again
                    if (csmacaCanProceed(0.0, true)) // Case E1
                    {
                        tmpCsmaca = 0;
                        csmacaCallback(phy_IDLE);
                        // slotted CSMA-CA successfully
                        return;
                    }
                    else // Case E2: postpone until next beacon
                        // sent or received
                    {
                        CW = 2;
                        bPeriodsLeft = 0;
                        csmacaWaitNextBeacon = true;
                        // Debugged
                    }
                }
                else handleBackoffTimer(); // Case D2: perform CCA
                //again, this function sends a CCA_request
            }
        }
        else // busy channel
        {
            if (beaconEnabled)
                CW = 2; // Case C2
            NB++;

            if (NB > mpib.macMaxCSMABackoffs) // Case F1
            {
                tmpCsmaca = 0;
                csmacaCallback(phy_BUSY);
                return;
            }
            else {
                BE++;
                if (BE > aMaxBE) BE = aMaxBE; // aMaxBE defined in
                // Ieee802157Const.h
                csmacaStart(false); // not the first time
            }
        }
    }
}

```

**Código 5.4.** Código para *Propuesta 03. Cambio 03* (continuación).

```

/*
Cambio para VLC:
/* * CAMBIOS IEEE802157: New handle_PLME_CCA_confirm of Random Access Algorithm.
* Three cases:
* Case 1: Channel idle, so transmit. POTENTIAL SUCCESS
* Case 2: Channel busy but no reaches maximum retrys yet.RETRY
* Case 3: Channel busy and more retrys are no possible. FAILURE
* CW method disappears.
* Same cases to slotted and unslotted.
**/
*/

void Ieee802157Mac::handle_PLME_CCA_confirm(PHYenum status)
{
    if (taskP.taskStatus(TP_CCA_CSMACA))
    {
        taskP.taskStatus(TP_CCA_CSMACA) = false
        // the following from Csmaca-CA802_15_7::CCA_confirm
        if (status == phy_IDLE) // idle channel
        {
            // CAMBIO IEEE802157: if Iddle then transmit.
            //potential success
            tmpCsmaca = NULL;
            csmacaCallback(phy_IDLE); //slotted and unslotted successfully,
            callback
            return;
        }
        else // busy channel, set NB and BE
        {
            NB++;
            BE++;
            if (BE > aMaxBE)
                BE = aMaxBE; // aMaxBE defined in Ieee802157Const.h
            if (NB mpib.macMaxRABackoffs) // new PIB attribute
                macMaxRABackoffs
            {
                //failure
                tmpCsmaca = 0;
                csmacaCallback(phy_BUSY);
                return;
            }
            else
            {
                //retry
                csmacaStart(false); // not the first time
            }
        }
    }
}

```

#### 5.7.2.3.4. Cambio 04.

Este último cambio es para la opción en que la red esté sin detección de portadora activa. En este caso, el cambio se produciría en el método “*Ieee802157Mac::csmaca\_handle\_RX\_ON\_confirm*” pues éste es quien se dispone a escuchar una vez recibe la confirmación de estado RX\_ON, si no está activa la detección retomará el proceso con el “*Ieee802157Mac::csmacaCallBack*” con la opción trucada de canal libre, para emitir “a ciegas”.

Este cambio se muestra en el **Código 5.5** y **5.6** al final del capítulo.

Código 5.5. Código para *Propuesta 03. Cambio 04*.

```

/*
 * Ieee802157Mac::csmaca_handle_RX_ON_confirm
 * Actual
 */

void Ieee802157Mac::csmaca_handle_RX_ON_confirm(PHYenum status)
{
    simtime_t now, wtime;
    if (status != phy_RX_ON)
    {
        if (status == phy_BUSY_TX)
        {
            EV << "[CSMA-CA]: device is busy Txing now, RX_ON will be set later" <<
endl;
            taskP.taskStatus(TP_RX_ON_CSMACA) = true;
            EV << "[MAC-TASK]: TP_RX_ON_CSMACA = true" << endl;
        }
        else
        {
            EV << "[CSMA-CA]: RX_ON request did not succeed, try again" <<
endl;
            handleBackoffTimer();
        }
        return;
    }

    // phy_RX_ON
    // locate backoff boundary if needed
    EV << "[CSMA-CA]: ok, device is set to RX_ON " << endl;
    now = simTime();
    if (beaconEnabled)
    {
        EV << "[CSMA-CA]: locate backoff boundary before sending CCA request" <<
endl;
        wtime = csmacaLocateBoundary(toParent(tmpCsmaca), 0.0);
    }
    else
        wtime = 0.0;
    if (wtime == 0.0)
    {
        taskP.taskStatus(TP_CCA_CSMACA) = true;
        EV << "[MAC-TASK]: TP_CCA_CSMACA = true" << endl;
        PLME_CCA_request();
    }
    else
        startDeferCCATimer(wtime);
}
}

```

Código 5.6. Código para *Propuesta 03. Cambio 04* (continuación).

```

/*
Cambio para VLC:
/* * CAMBIOS IEEE802157: New csmaca_handle_RX_ON_confirm of Random Access Algorithm.
* If CCA is not supported, transmit whithout PLME CCA request,
* via a Callback with PHY status tricked.
* Blind transmission included**/
*/

void Ieee802157Mac::csmaca_handle_RX_ON_confirm(PHYenum status)
{
    simtime_t now, wtime;
    if (status != phy_RX_ON)
    {
        if (status == phy_BUSY_TX)
        {
            EV << "[CSMA-CA]: device is busy Txing now, RX_ON will be set later" <
            < endl;
            taskP.taskStatus(TP_RX_ON_CSMACA) = true;
            EV << "[MAC-TASK]: TP_RX_ON_CSMACA = true" << endl;
        }
        else
        {
            EV << "[CSMA-CA]: RX_ON request did not succeed, try again" << endl;
            handleBackoffTimer();
        }
        return;
    }
    // phy_RX_ON
    // locate backoff boundary if needed
    EV << "[CSMA-CA]: ok, device is set to RX_ON " << endl;
    now = simTime();
    if (beaconEnabled)
    {
        EV << "[CSMA-CA]: locate backoff boundary before sending CCA request" <<
endl;
        wtime = csmacaLocateBoundary(toParent(tmpCsmaca), 0.0);
    }
    else
        wtime = 0.0;
    if (wtime == 0.0)
    {
        if (tmpCsmaca.getisCCAsupport())
        {
            taskP.taskStatus(TP_CCA_CSMACA) = true;
            EV << "[MAC-TASK]: TP_CCA_CSMACA = true" << endl;
            PLME_CCA_request();
        }
        else
        {
            EV << "[CSMA-CA]: Carrier sense not active. BLIND Transmission" <<
endl;
            tmpCsmaca = NULL;
            csmacaCallBack(phy_Idle); //POTENTIAL SUCCESS
        }
    }
}

```

### 5.7.2.3.5. Cambio 05.

Este último cambio afecta a los reintentos de confirmación, se encuentra aún bajo revisión de funcionamiento.

## 5.8. Modo indirecto

También se hizo necesario implementar el manejo de transacción de datos por modo indirecto. En el **Anexo IV** se describe el funcionamiento de manejo de transacción según el estándar IEEE802.15.7.

A continuación se describe las acciones que debe ejecutar el código para la simulación de una transmisión de datos en modo indirecto, separando la rutina del coordinador y los dispositivos asociados, y que fueron implementadas en este PFC.

### 5.8.1. Rutina del coordinador para una transacción en modo indirecto

Al recibir un dato de la capa superior del tipo:

```
[(cSimpleModule)handleMessage-> handleUpperMessage-> MCPS_DATA_request],
```

en el caso de que el coordinador esté en modo de transmisión indirecto [`txOption = 2` (INDIRECT\_TRANS)] (actualmente cada host sólo puede estar en un modo), deberá actualizar la variable temporal que guarda los campos de direcciones pendientes de la siguiente beacon, así como almacenar la trama de datos recibida y eliminará la tarea pendiente creada al comienzo del método `MCPS_DATA_request`, pues ésta se crea para el caso de modo directo.

El número de tramas pendientes que se pueden almacenar son 7 y se guardará las direcciones de destino según el orden de llegada de cada trama. Para almacenar las direcciones de dichas tramas, se utilizará una lista anidada, lo que permitirá actualizar, comprobar duplicación de tramas, eliminar e insertar nuevas fácilmente y en orden de llegada.

El tiempo para almacenar una trama es el que indique el atributo `MAC macTransactionPersistenceTime`, por tanto si se almacena correctamente la trama le corresponderá un evento con el nuevo tipo de temporizador «`trPersistenceTimer`». Para poder controlar con un sólo temporizador hasta 7 timeouts distintos por trama pendiente almacenada, se añadirá un campo de «`timeout`» a los nodos de la lista anidada.

Si el coordinador recibe un mensaje del tipo «`trPersistenceTimer`» borra el primer nodo de la lista y planifica el siguiente evento, en el caso de haber más tramas pendientes almacenadas, con el valor correspondiente de «`timeout`» también guardado.

Si el coordinador, antes de dicho tiempo, recibe un comando de tipo `Ieee802157_DATA_REQUEST`:

```
[(cSimpleModule)handleMessage-> handleLowerMessage-> handleCommand-> handle_data_request_cmd (nuevo)],
```

buscará en la lista si existe la trama correspondiente, es decir, comprobará que existe un nodo con dicha dirección y enviará una trama ACK con la información correspondiente, es decir, si hay dato pendiente para el dispositivo del que ha recibido el comando de solicitud o no existe dato pendiente para tal.

Si existe dicha trama pendiente con esa dirección de host de destino, guardará el puntero en «*txData*» e iniciará el proceso de tarea pendiente TP\_DATA\_REQUEST con su correspondiente algoritmo para el caso de modo indirecto [*txOption* = 2·(INDIRECT\_TRANS)], y que coordinará el resto del proceso.

Este proceso, de forma global, consistirá en primer lugar, en comprobar si el proceso completo entra dentro del CAP de la supertrama actual.

Si no es así, el coordinador descartará la trama de «*txData*», y finalizará la tarea con fallo (repetirá el proceso si vuelve a recibir el comando de solicitud del dispositivo).

Si la transmisión se puede completar en el CAP de la supertrama actual, solicitará el modo transmisión en el transceptor. Si el PHY no responde a dicha petición con éxito, la tarea finalizará con fallo. Si el nivel PHY responde a dicha petición con éxito, se transmitirá el dato (*txPkt* = *txData*) inmediatamente (*sendDown*), y se esperará confirmación si la trama así lo especifica durante *mac\_AckWaitTime\_seg*.

Si en cambio, no requiere confirmación la tarea finalizará con éxito. Si requiere confirmación y el coordinador recibe el ACK en el tiempo permitido, la tarea finaliza con éxito. En los casos en los que ocurra que se cumpla el tiempo de espera de confirmación la tarea finalizará fallida. Cuando la tarea finaliza con éxito, se elimina el nodo correspondiente a la trama de la lista anidada de tramas pendientes.

En cualquier otro caso, como este modo no permite reintentos, será el dispositivo el que volverá a solicitar el dato si no lo recibe en el tiempo en el que está en modo recepción, y si aún lo ve pendiente en la siguiente baliza, lo cual dependerá del *macTransactionPersistenceTimer*.

También existe el caso en que la lista está llena, que involucra la primitiva MCPS\_DATA\_confirm con estado TRANSACTION\_OVERFLOW, o simplemente tener en cuenta el límite de la lista cuando llegue una solicitud de dato en modo indirecto. Además se produce la llamada al método cuando se activa TRANSACTION\_EXPIRED.

### 5.8.2. Rutina del dispositivo para solicitar un dato en una transacción en modo indirecto

En la recepción de una baliza, el dispositivo asociado, comprobará si su dirección está en el subcampo lista de direcciones pendientes de dicha trama. En el caso de estar su dirección en la lista, iniciará el proceso de envío de la trama de comando del tipo Ieee802157\_DATA\_REQUEST, este proceso se realizará mediante el control de una tarea pendiente.

Dentro de la tarea, se pondrá en modo transmisión, una vez en dicho modo pasará al algoritmo para tratar de enviar la trama de solicitud por medio del algoritmo de acceso al medio (dentro del cual se comprueba que el envío de la trama y la recepción de su confirmación entran en el periodo de contención), luego se pondrá en modo recepción y pasará a la espera de un ACK.

Si se recibe un ACK antes de finalizar el CAP de la supertrama actual, se ha de comprobar si el subcampo de trama pendiente del ACK está a uno, en cuyo caso, el coordinador le estará confirmando que efectivamente existe una trama pendiente para él y el dispositivo se pondrá en modo recepción inicializando un nuevo evento para el temporizador *macMaxFrameTotalWaitTime*. Si el subcampo de trama pendiente del ACK está a cero, el coordinador le estará indicando que no hay trama pendiente, y se

finalizará la tarea. En caso de no recibir el ACK antes de finalizar el tiempo *macMaxFrameTotalWaitTime*, se finalizará la tarea pendiente con fallo.

Estas rutinas son algo más complejas, e incluyen unos estados mejor definidos.

### **5.8.3. Comando de solicitud de dato**

Si el comando de solicitud de dato está siendo enviado como respuesta de la recepción de una trama de baliza que indica que hay dato pendiente para el dispositivo, el subcampo modo de direccionamiento de destino del campo de control de trama debe establecerse a cero si el campo de especificación de supertrama indicado en su trama de baliza

### **5.8.4. Algoritmo para el comando de solicitud de dato en el dispositivo**

Se enviarán comandos entre dispositivos tal, lo que implicará un nuevo método como puente a la actual transmisión en CAP con CSMACA pero para un comando.

### **5.8.5. Algoritmo de gestión para la recepción de un comando de solicitud de dato en el dispositivo**

Consiste un método para dicha gestión.

## **5.9. Conclusiones**

En este capítulo se describe la metodología de desarrollo empleada durante el desarrollo del módulo MAC en la herramienta OMNeT++. Se realiza una descripción de la estructura global del simulador, de sus módulos constituyentes principales y del funcionamiento de cada uno de ellos. Se presta mayor dedicación y detalle en el módulo referente a la capa de red y MAC, que es el objetivo principal del proyecto y donde se deben lograr implementar las especificaciones del estándar.



# Capítulo 6

## Simulaciones y análisis

---

"La felicidad es saber unir el final con el principio."

PITÁGORAS

### 6.1. Introducción

En este capítulo se describen los tests que se diseñaron y ejecutaron adecuados para la verificación de la funcionalidad del módulo MAC implementado. Una vez descritas las pruebas y las simulaciones que fueron efectuadas, se mostrarán y analizarán los resultados para un escenario simple de dos nodos. Los resultados permiten validar el funcionamiento del nivel de acceso al medio y verificar que cumple las especificaciones del estándar 802.15.7.

### 6.2. Parametrización de las simulaciones

El modelo de red VLC implementado está compuesto por una serie de unidades funcionales bien diferenciadas, como se deduce de la descripción funcional. Esas unidades funcionales, típicas de una arquitectura de comunicaciones, se materializan en módulos componentes, típicos de cualquier modelo de simulaciones de OMNET++.

Las simulaciones de tráfico se encuentran en un directorio paralelo al código fuente que construye y simula el comportamiento de la red. En ella es donde se describe una red de topología en estrella mediante el lenguaje propio de OMNET++, así como su organización. Esta descripción la componen, desde módulos simples, hasta archivos de configuración de escenarios *.ini*, en los que se define el comportamiento de los módulos y se informa de los valores de los parámetros de red variables descritos en el proyecto de red VLC estrella. Los parámetros pueden definir el número de submódulos, número de puertas, y la forma que tienen las conexiones internas, de forma que se construye la topología de manera flexible.

Vemos en la **Tabla 6.1** los submódulos principales de nuestros escenarios de red y su descripción. Atenderemos a los parámetros de la **Tabla 6.2** que fueron los más influyen en las variaciones necesarias para el objetivo de estudio de este PFC. Cada uno de los parámetros está instanciado en su correspondiente archivo *.ned* cada uno de ellos iniciados con un valor por defecto, y los archivos de configuración permiten modificar dicho valor de aquellos parámetros que convengan, de cada parte de la red, para la definición de cada escenario.

La mayoría de las parametrizaciones se hicieron en la interfaz de red del módulo compuesto de nodos, aprovechando las funcionalidades ya definidas en el resto. Con los archivos de simulación construimos distintos grupos de configuración orientados a los distintos objetivos del proyecto, principalmente los dividimos en:

- Simulaciones de **comprobación**: Esperamos el mismo resultado antes y después del cambio.

- Simulaciones de **validación**: Esperamos resultados que atiendan a una funcionalidad nuevamente implementada.
- Simulaciones de **comparación y predicción**: Recopilación de resultados bajo distintas variaciones de parámetros de simulación. Depende de parámetros estadísticos.

Tabla 6.1 Módulos para la topología en estrella de red VLC, *Ieee802157StarNet*.

Módulo	Descripción
<i>Hosts / Dispositivos ópticos, Ieee802157Node</i>	Módulo compuesto que define los componentes de un nodo. Implementa un nodo simplificado para red IEEE 802.15.7 (con capa PHY y MAC conforme a esa norma), más unos módulos sencillos de red y aplicación para hacer pruebas, y módulos auxiliares para batería, notificaciones y movilidad. // @author Grupo VLC. DIT ULPGC. (Basado en INETMANET).
<i>channelControl / Control del medio de transmisión, Ieee802157ChannelControl</i>	Módulo simple que tiene exactamente una instancia en cada modelo de red que contenga nodos móviles e inalámbricos. Este módulo es el encargado de gestionar el medio de transmisión para posibilitar simular el comportamiento del medio. Para ello se mantiene informado de la localización y el movimiento de todos los nodos de la red, determinando que nodos están en comunicación (por debajo de la máxima distancia de interferencia de la red). Esta información de posición y orientación en cada momento, es usada por los módulos PHY de los nodos cuando quieren transmitir. // @author Grupo VLC. DIT ULPGC (Basado en INETMANET).
<i>Annotation / Control de anotaciones, Ieee802157Anotation</i>	Módulo simple para el manejo de las anotaciones. Las anotaciones son líneas y polígonos, dibujados sobre el entorno gráfico de simulación a fin de representar alguna información útil (p. ej: la orientación de los nodos - dispositivos ópticos de la red). Cuando está presente en una red, se encarga de gestionar las anotaciones que se pretende que aparezcan en el entorno gráfico a simular. // @author Grupo VLC. DIT ULPGC (Basado en INETMANET).

Tabla 6.2 Parámetros principales para la configuración de escenarios.

Parámetro de capa de red	Descripción
Coordinador VPAN	A cada nodo se le asigna este valor falso o verdadero. De esta forma se define fácilmente la topología en estrella, asignando sólo como nodo coordinador al host[0], y falso a cualquier otro.
Parámetro de capa de aplicación (generación de tráfico)	Descripción
Tamaño de paquetes	Especifica el tamaño de los paquetes de datos.
Intervalo entre paquetes transmitidos	Especifica el tiempo entre solicitud de transmisión de paquetes.
Instante de primer paquete	Especifica el tiempo de la primera transmisión de la simulación.
Nodo recolector	En modo no recolector depende de la capa física que se envíen los mensajes y no recibe ACK, si el valor es verdadero,
Destino del tráfico	Cadena con el identificador del nodo al que la capa de aplicación.

Tabla 6.2 Parámetros principales para la configuración de escenarios (*continuación*).

Parámetro de capa MAC	Descripción
BO, disposición de baliza	Factor de configuración de la duración de supertrama
SO, disposición de supertrama	Factor de configuración del intervalo entre balizas.
Ventana de contención	Opción seleccionada para el mecanismo de retardo aleatorio.
CCA, exploración del canal	Activa el mecanismo CCA en el proceso de transmisión en modo directo de los dispositivos.
Recepción de GTS	En caso de que el modo seleccionado sea transmisión GTS, con el valor true, el dispositivo receptor podrá recibir en modo GTS.
Parámetro de capa PHY	Descripción
MCS ID de transmisión	Modo seleccionado de modulación y codificación para la tasa de reloj de transmisión de los dispositivos.
MCS ID de recepción	Modo seleccionado de modulación y codificación para la tasa de reloj de recepción de los dispositivos.
Máxima frecuencia de reloj de transmisión	Máximo reloj soportado por el transmisor óptico.
Máxima frecuencia de reloj de recepción	Máximo reloj soportado por el receptor óptico.
Selector de tasa de reloj	Define el MCS ID (modulación y codificación escogida) en el que los dispositivos transmiten en el periodo de acceso de contención de subida.

### 6.3. Parametrización de las estadísticas de resultados

Para la recopilación de estadísticas, tal como se explica en el capítulo de la herramienta OMNET++, se instancian en el método de inicialización de cada módulo de nivel de acceso al medio para cada nodo, las variables del sistema que luego se presentan en pantalla y en tiempo de ejecución. Esto se consigue utilizando la macro de OMNET *WATCH()* que las convierte en parámetros observables durante la simulación. Los valores recogidos durante las simulaciones de comparación y predicción fueron los numerados en la **Tabla 6.3**.

Tabla 6.3 Variables de sistema para recopilación de estadísticas.

Variable	Descripción
<i>bcnLossCounter</i>	No. de balizas perdidas.
<i>inTxSD_txSDTimer</i>	En modo de transmisión activo.
<i>inRxSD_rxSDTimer</i>	En modo de recepción activo.
<i>numUpperPkt</i>	No. de paquetes transmitidos en tiempo de subida.
<i>numUpperPktLost</i>	No. de paquetes transmitidos en tiempo de subida y perdidos.
<i>numUpperPktRequested</i>	No. de paquetes solicitados en tiempo de subida.
<i>numCollision</i>	No. de colisiones.
<i>numFramesRxDuringCCA</i>	No. de tramas recibidas durante la exploración del canal de transmisión en el nodo.
<i>numLostBcn</i>	No. total de balizas perdidas.
<i>numTxBcnPkt</i>	No. total de balizas transmitidas desde el nodo.
<i>numTxDataSucc</i>	No. de transmisiones con éxito de paquetes de datos desde el nodo.
<i>numTxDataFail</i>	No. de transmisiones fallidas de paquetes de datos desde el nodo.
<i>numTxGTSSucc</i>	No. de transmisiones con éxito en modo GTS desde el nodo.

Tabla 6.3 Variables de sistema para recopilación de estadísticas (*continuación*).

Variable	Descripción
<i>numTxGTSTFail</i>	No. de transmisiones fallidas en modo GTS desde el nodo.
<i>numTxAckPkt</i>	No. de paquetes de confirmación transmitidos desde el nodo.
<i>numRxBcnPkt</i>	No. de paquetes de baliza recibidos en el nodo.
<i>numRxDataPkt</i>	No. de paquetes de datos recibidos en el nodo.
<i>numRxDupDataPkt</i>	No. de paquetes de datos duplicados recibidos en el nodo.
<i>numRxGTSPkt</i>	No. total de paquetes recibidos en modo GTS en el nodo.
<i>numRxAckPkt</i>	No. de paquetes de confirmación recibidos en modo GTS en el nodo.
<i>numTxAckInactive</i>	No. de paquetes de confirmación pendientes en periodo activo de la supertrama en el nodo.
<i>numTxDataPktAttempts</i>	No. total de intentos de transmisión de paquetes de datos desde el nodo.
<i>numCSMAChannelAccess-Failures</i>	No. de fallos de acceso al canal mediante el mecanismo CSMA del nodo.
<i>numCCAIdleResults</i>	No. de resultados de exploración del canal libre.

Dependiendo del tipo de prueba realizado en cada fase, fueron unas u otras las variables que nos aportaron información útil. Por este motivo, comentaremos los resultados recogidos de las simulaciones aplicadas tras cada fase de desarrollo por separado.

Tanto en el caso de acondicionamiento como en los mecanismos posteriormente implementados, la fase de desarrollo y pruebas es paralela en ejecución, aunque para facilitar la comprensión la hemos dividido en apartados independientes. Por tanto para cada fase de implementación, leeremos una pequeña introducción sobre el objetivo a desarrollar, la fase de desarrollo, la fase de simulación y comprobación, y por último las conclusiones y discusión de resultados.

#### 6.4. Simulaciones de comprobación

Durante esta fase, las pruebas se limitaron a simulaciones de comprobación. Éstas sirvieron a lo largo del proyecto como referencia para la validación de nuevas funcionalidades del protocolo VLC sin interferir en el resto de mecanismos no modificados.

Para la comprobación de los continuos cambios se eligió un escenario, *BasicEscenario2\_nodos*, con las siguientes características:

- Topología en estrella
- Consta de dos nodos: un coordinador estacionario y un dispositivo móvil. El coordinador está centrado en la habitación y el dispositivo móvil se configura para que permanezca estacionario también. La distancia entre nodos de 1.5m.
- Los nodos están perfectamente alineados entre sí, lo cual evita que la potencia recibida del modelo de visión directa (LOS, *Line of Sight*) de la capa PHY afecte en los valores obtenidos.

### 6.4.1. Escenario *BasicEscenario2\_nodos*

En este primer escenario controlamos la secuencia de transmisión de tramas de baliza del nodo coordinador a otro nodo que transmite tramas de datos en modo directo desde la capa de aplicación y recibe las tramas de confirmación correspondientes. En las **Figura 6.1** y **6.2** vemos como, con los *logs* y la interfaz gráfica del OMNET++ para analizar la simulación, comprobamos dicho funcionamiento entre dos nodos.



Figura 6.1. Transmisión de baliza de supertrama.

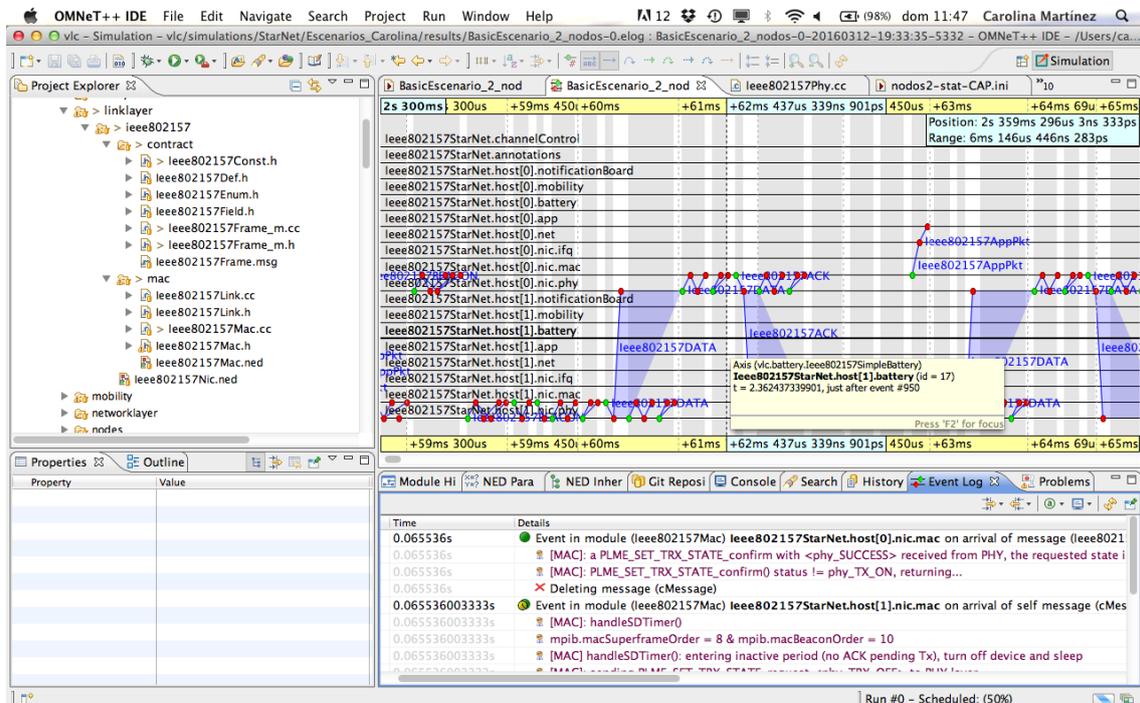


Figura 6.2. Transmisión de datos en modo directo.

Inicialmente, la configuración escogida para las pruebas de comprobación se definió con los valores recogidos en la **Tabla 6.4** y **6.5**. La diferencia es que se consideraba evaluar valores de *BO* y *SO* que suponían las siguientes constantes para los temporizadores de instante de inicio de periodo de transmisión de supertrama e instante de transmisión de baliza se resumen en la **Tabla 6.6**.

**Tabla 6.4** Configuración característica inicial de los parámetros de definición de supertrama.

Parámetro	Valor	Descripción
<i>numHost</i>	2	La red la forman dos dispositivos.
<i>host[0].app.isSink</i>	true	El dispositivo <i>host[0]</i> no envía datos de la capa de aplicación.
<i>host[1].app.isSink</i>	false	El dispositivo <i>host[1]</i> envía datos de la capa de aplicación.
<i>packetSize</i>	10B	Los paquetes de datos enviados son del tamaño 10Bytes.
<i>interDepartureTime</i>	0.1s	El tiempo entre orden de intento de transmisión de nuevo paquete de datos es de 0.1s, tiempo de simulación.
<i>firstPacketTime</i>	1s	El primer intento de transmisión de paquete de datos ocurrirá en el tiempo de simulación 1 segundo.
<i>BO</i> , Orden de baliza	$\$BO = 10..11$	El orden de baliza de super-trama se simulará para los valores 10 y 11.
<i>SO</i> , Orden de super-trama	$\$SO = (BO - 2)..(BO)$	El orden de super-trama se simulará para para dos valores inferiores para cada valor de <i>BO</i> .

**Tabla 6.5** Configuración característica inicial de los parámetros de definición de supertrama.

Parámetro	Valor	Descripción
<i>BO</i> , Orden de baliza	$\$BO = 10..11$	El orden de baliza de super-trama se simulará para los valores 10 y 11.
<i>SO</i> , Orden de super-trama	$\$SO = (BO - 2)..(BO)$	El orden de super-trama se simulará para para dos valores inferiores para cada valor de <i>BO</i>

**Tabla 6.6** Valores iniciales de supertrama de simulación.

Parámetro	Duración de Supertrama	Tiempo entre balizas
<i>BO = 10, SO=8, CLK = 200 KHz</i>	1228,8 ms	4915,2 ms
<i>BO = 10, SO=9, CLK = 200 KHz</i>	2457,6 ms	4915,2 ms
<i>BO = 10, SO=10, CLK = 200 KHz</i>	4915,2 ms	4915,2 ms
<i>BO = 11, SO=9, CLK = 200 KHz</i>	2457,6 ms	9830,4 ms
<i>BO = 11, SO=10, CLK = 200 KHz</i>	4915,2 ms	9830,4 ms
<i>BO = 11, SO=11, CLK = 200 KHz</i>	9830,4 ms	9830,4 ms

Para el resto de niveles mantuvimos una configuración estándar para estas pruebas de comprobación. A partir de esta parametrización, la cual tiene un nodo coordinador y un nodo transmisor de datos, se pretendía controlar el envío de tramas de baliza, es decir, verificar que se mantiene el mecanismo de supertrama para la comunicación entre nodos

en la red (Figura 6.3.). Se varían únicamente los parámetros referentes a la diferencia de tiempo de intento de transmisión de paquetes de datos. De esta forma se validaron los cálculos temporales de la implementación, que debían conservarse tras los cambios. Además nos permitió encontrar las primeras carencias y errores del código desarrollado en INETMANET para WPAN heredadas.

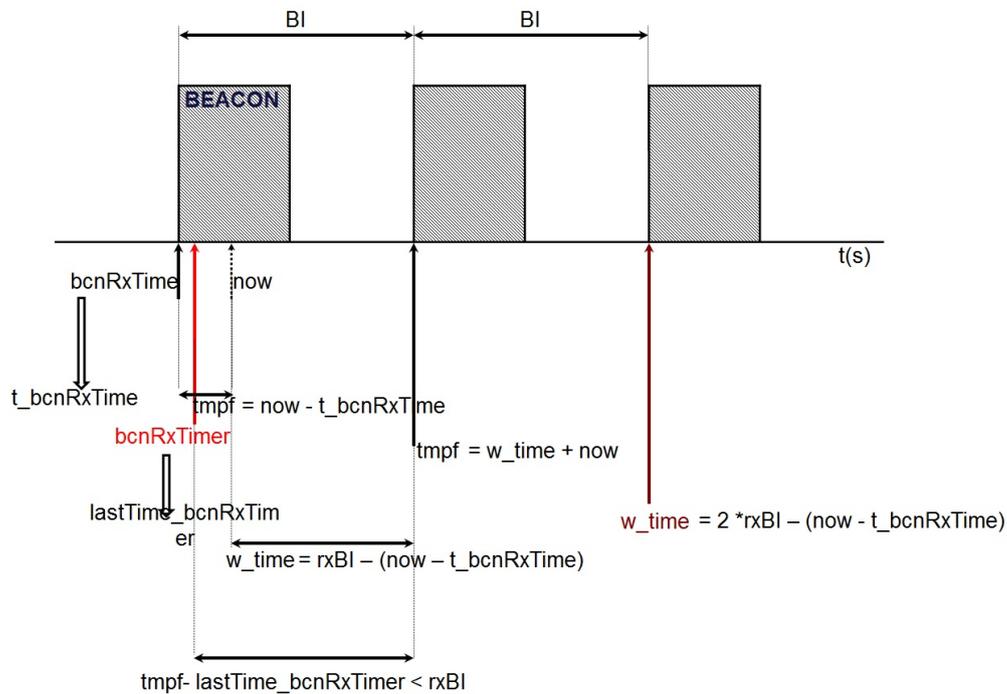


Figura 6.3. Supertrama

Con dicha configuración obtenemos cuatro ficheros de resultados de estadísticas por cada simulación, dos valores de  $SO$  por cada valor de  $BO$  10 y 11.

#### 6.4.2. Resultados y discusión de resultados

Los primeros resultados, además de las proporcionaron información sobre los puntos de colección de estadísticas escalares en el sistema, y sobre las prestaciones del entorno de trabajo INETMANET. En general conclusiones que se describen a continuación en el apartado, nos, una visión de los puntos dónde se encontraban las carencias identificadas en el estudio previo a las fases de implementación.

Los valores recogidos durante la simulación nos permitieron llegar a las siguientes conclusiones iniciales sobre el estado del nivel MAC del que partíamos:

- **Estado del coordinador:** Los métodos propios del nodo configurado como coordinador, están implementados en la misma clase C++ que el resto de métodos propios del nivel MAC. Esto dificultó trazar el comportamiento, principalmente en lo que se refiere a variables globales que contienen valores temporales, y los cuales son condición para el arranque de los distintos eventos de la ejecución.
- **Colisión de paquetes:** en este escenario se detectaron por primera vez.

- **Modo GTS:** La implementación de este modo es una aproximación a la especificación del estándar. El descriptor GTS, que debería estar en las tramas de baliza, se encuentra sin embargo en un método del nivel MAC o clase *Ieee802157Mac*. El modo GTS podría ser parte de una posible solución para el problema del nodo oculto en estas redes. Por tanto, se crearon una serie de archivos de simulación específicos para estudiar el comportamiento y los resultados obtenidos con esta solución existente, así como la posibilidad de crear el comando de petición de GTS e implementar el modo según la especificación.
- **Modo con baliza habilitada:** además de algunos errores de cálculos temporales del modo de transmisión directo, detectamos una carencia que nos guió a la siguiente fase del desarrollo, descrita en la siguiente sección. Para completar el mecanismo del modo de transmisión con baliza habilitada debía permitir el modo en que el orden de baliza y el orden de supertrama coincidieran en valor, y por tanto un modo único entre supertramas de periodos *CAP*.

La tercera y sexta opción de la tabla equivale a una supertrama sin modo en espera o enteramente con periodos *CAP*, con cuya parametrización se encontró que la simulación resultaba fallida y se consideró en la siguiente fase de implementación. Para el resto de opciones de configuración, los resultados de rendimiento eran los siguientes.

### 6.5. Modo con baliza habilitada

Durante el análisis de resultados de las simulaciones de comprobación, se detecta que para el modo de transmisión con baliza habilitada la configuración de supertrama con el valor del orden de supertrama igual al valor de orden de baliza provoca un error.

En la **Figura 6.4** se muestra el caso simulado para  $SO=BO$ , ocurre que coinciden los eventos correspondientes a final de supertrama, y transmisión de la siguiente baliza, *txSDTimer* y *bcnTxTimer*, instante capturado en la imagen de simulación mostrada.

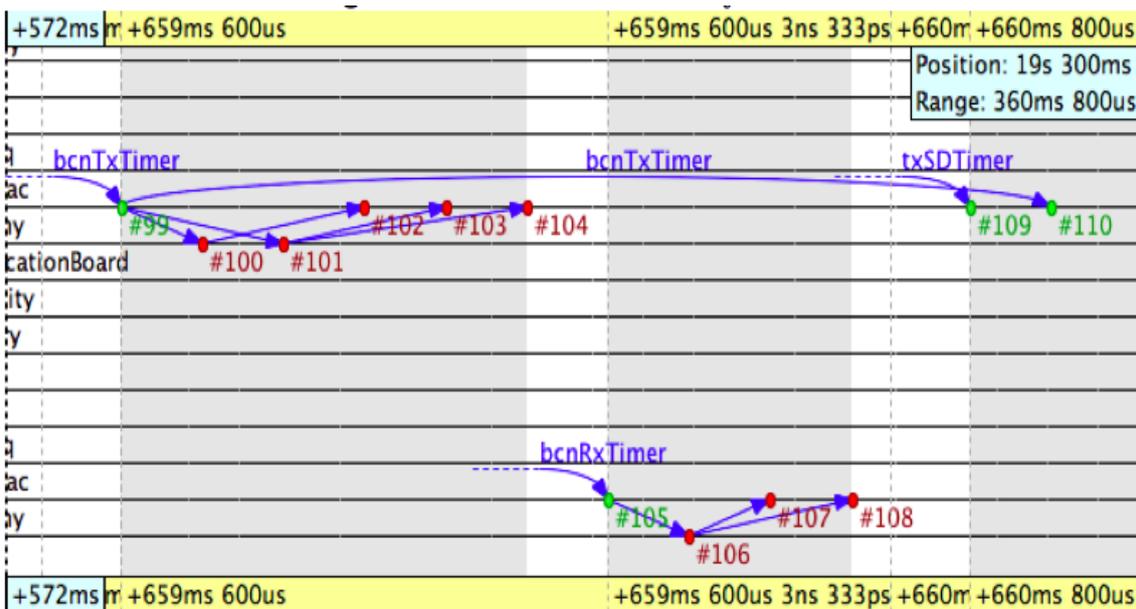


Figura 6.4. Eventos *txSDTimer* y *bcnTxTimer* en la simulación inicial.

En la simulación se ha detectado que estos eventos *txSDTimer* y *bcnTxTimer*, son incompatibles pues al finalizar el tiempo activo de supertrama, se le indica a la capa física que apague el transceptor para entrar en modo *sleep*, y a la vez, es el momento de transmitir la siguiente baliza, lo que la capa física descarta por estar en dicho modo de suspensión. Este comportamiento se observa en la captura mostrada en la **Figura 6.5**.

```
[PHY]: a PLME_SET_TRX_STATE_confirm with 7 to MAC
phy_SUCCESS[PHY]: sending a PLME_SET_TRX_STATE_confirm with 7 to MAC
** Event #181 T=31.45728 Ieee802157StarNet.host[0].nic.phy (Ieee802157Phy, id=14), on 'Ieee802157BEACON' (Ieee802157BeaconFrame, id=143)
[PHY]: a MAC frame Ieee802157BEACON received from MAC layer but transceiver is disconnected, delete packet
** Event #182 T=31.45728 Ieee802157StarNet.host[0].nic.mac (Ieee802157Mac, id=13), on '{}' (Ieee802157MacPhyPrimitives, id=145)
[MAC]: a PLME_SET_TRX_STATE_confirm with 8 received from PHY, the requested state is 8
```

**Figura 6.5.** Comportamiento de la capa PHY tras recibir trama *Ieee802157BEACON* de transmisión.

Para solucionar el problema de baliza activada, se describieron en el **apartado 5.6** dos soluciones. La segunda solución, deja la posibilidad de que llegue al final de supertrama con *txACK* distinto de NULL controlada con un mensaje de error, e independiza la condición del modo *sleep* de ello.

La segunda solución fue adoptada para su implementación y se presenta la simulación aplicándola al escenario *BasicEscenario2\_nodos*.

### 6.5.1. Simulación de solución #2

Una vez tratada la posibilidad de la igualdad de *SO* y *BO* en el código, se crean distintos escenarios para comprobar su funcionalidad y el rendimiento de los dispositivos cuando varía *SO*.

Para ello se configura una frecuencia de transmisión de paquetes lo suficientemente grande para exista mucho tráfico en una misma supertrama, y así poder comparar distintos tamaños de ésta. Además, se consideran para dichos tamaños, cuatro valores de *SO* –8, 9, 10 y 11–, éste último igual al valor de *BO*. Este escenario sólo contempla transmisión directa entre dos nodos, un coordinador y un dispositivo.

Una vez obtenidos los resultados de las simulaciones, se observa que aunque el parámetro *goodput* debería aumentar según aumente el tamaño de supertrama, es menor en el caso de *SO* = 11. Los resultados se muestran en la **Figura 6.6**, **Figura 6.7** y **Figura 6.8**.

Esto es fácilmente corregido, pues el error está en las dimensiones de la variable que guarda los periodos de CAP restantes, el cual excedía las dimensiones de la unidad, convirtiendo el valor calculado en un valor erróneo.

Una vez corregido esto, se observa que tanto el rendimiento *goodPut* (Bytes/s) como el retardo entre extremos mejoran considerablemente según aumentamos el parámetro *SO*.

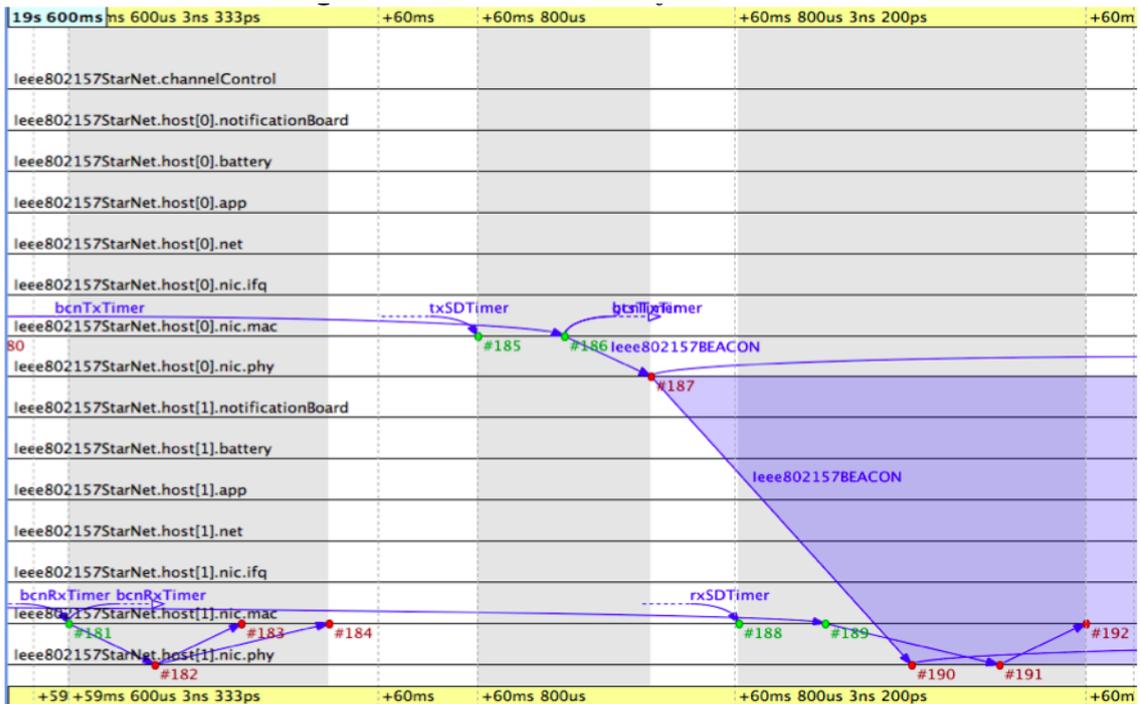


Figura 6.6. Resultado de simulación con solución #2. Eventos *txSDTimer* y *bcnTxTimer* en la simulación final.

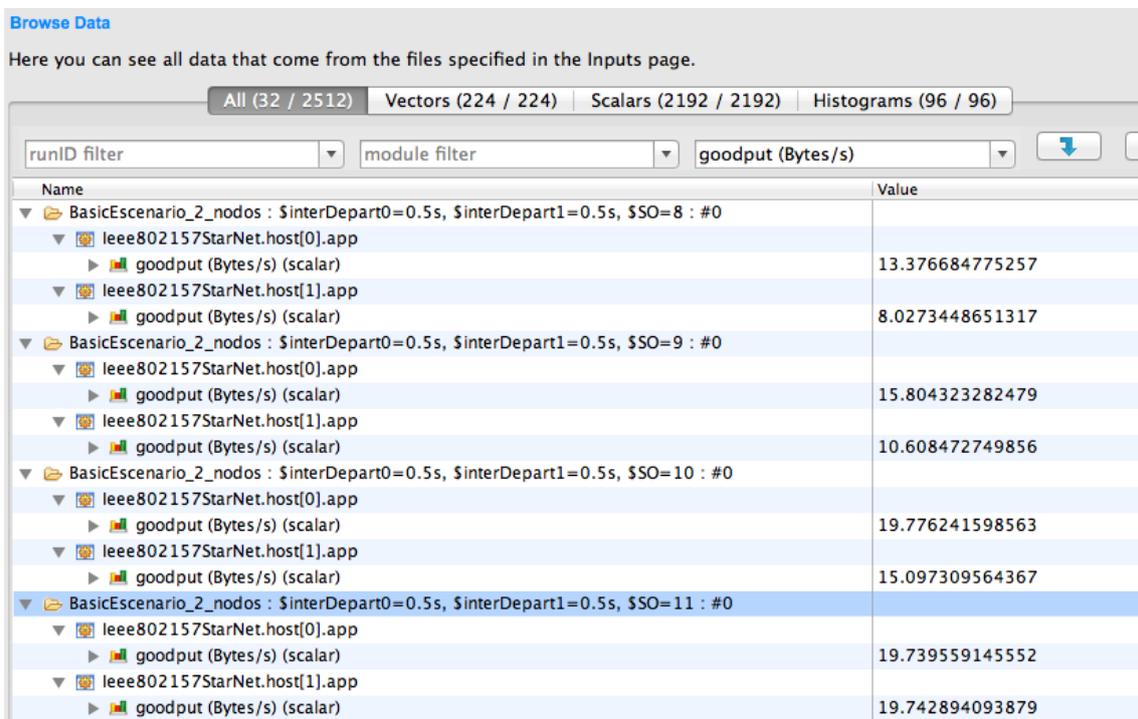


Figura 6.7. Resultado de simulación con solución #2. Detalle del parámetro “goodput”.

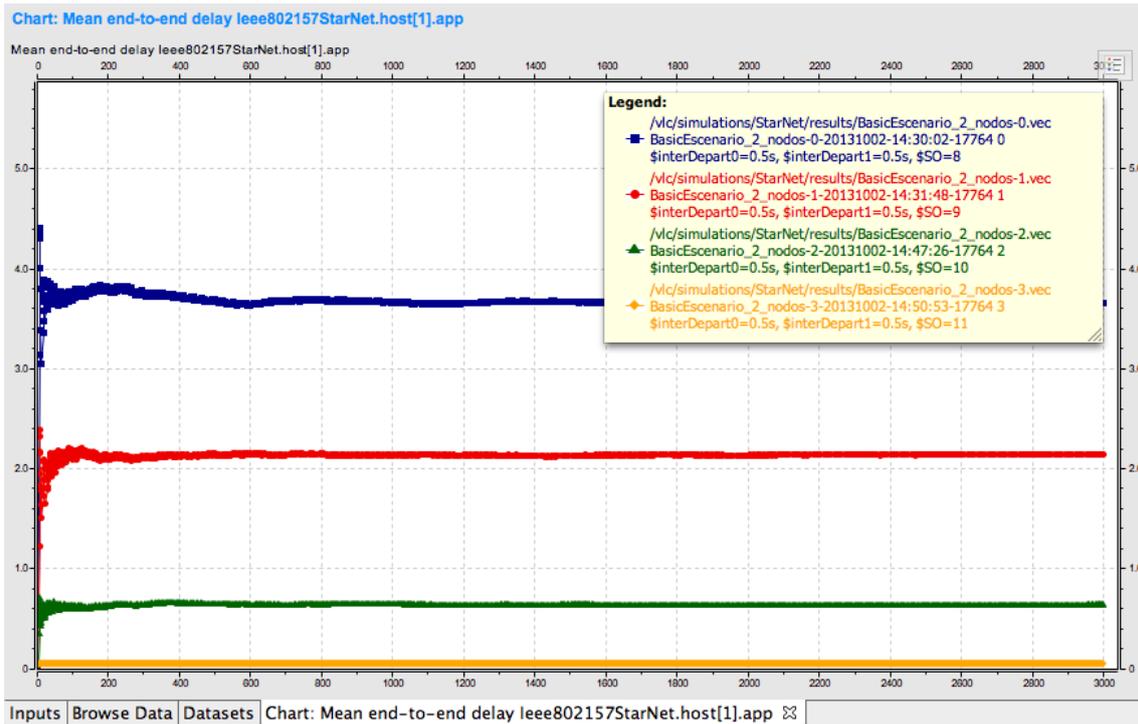


Figura 6.8. Resultado de simulación con solución #2. Detalle del parámetro “Mean end-to-end delay”.

### 6.5.2. Pérdida de sincronización.

Se realiza la simulación para observar el problema de la pérdida de sincronización respecto a las tramas beacon transmitidas por el coordinador. Esta disparidad a la hora de realizar dichos cálculos temporales, provoca una diferencia entre la recepción de baliza y la espera de dicha recepción en los dispositivos, debido al redondeo y truncamiento.

Simularemos el caso 1 de los errores de cálculo, en el cual el tiempo de recepción de una trama beacon en los dispositivos no corresponde con la recepción de ésta.

#### 6.5.2.1. Simulación del Caso #1

Los métodos asociados al proceso CSMA/CA, realizan los cálculos con el valor de recepción de baliza, por lo que no es problema el error de planificación del evento de recepción de siguiente beacon.

Las Figura 6.9 y 6.10 ilustran esta problemática.

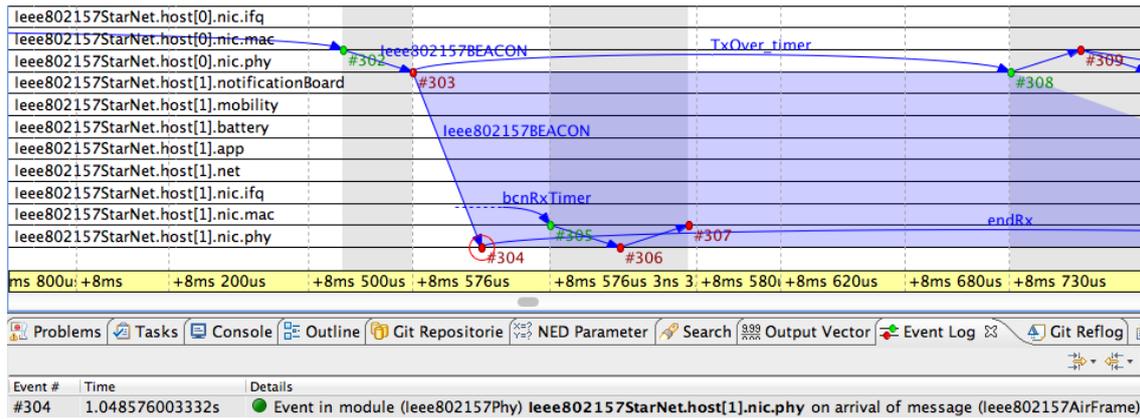


Figura 6.9. Instante real de recepción del beacon (1,048576003332).

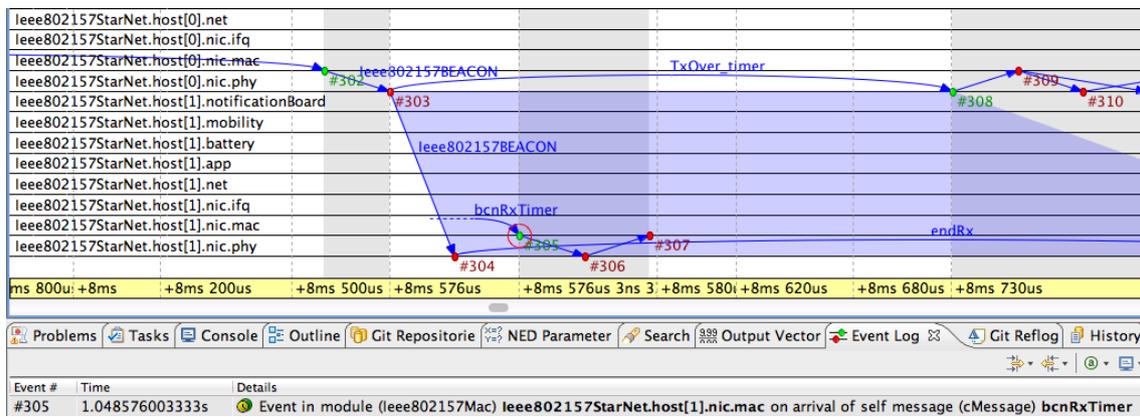


Figura 6.10. Instante programado para la recepción del beacon (1,048576003333).

## 6.6. Modo indirecto

Se ha lanzado la simulación para probar el algoritmo de gestión para la recepción de un comando de solicitud de dato en el dispositivo para el escenario *BasicEscenario2\_nodos*.

En la **Figura 6.11** y **6.12** se muestra la configuración para la prueba del modo indirecto y el resultado de la simulación para el caso de una transacción indirecta iniciada por el coordinador. En la **Figura 6.13** y **6.14** el mismo caso pero cuando la lista de transacciones está llena.

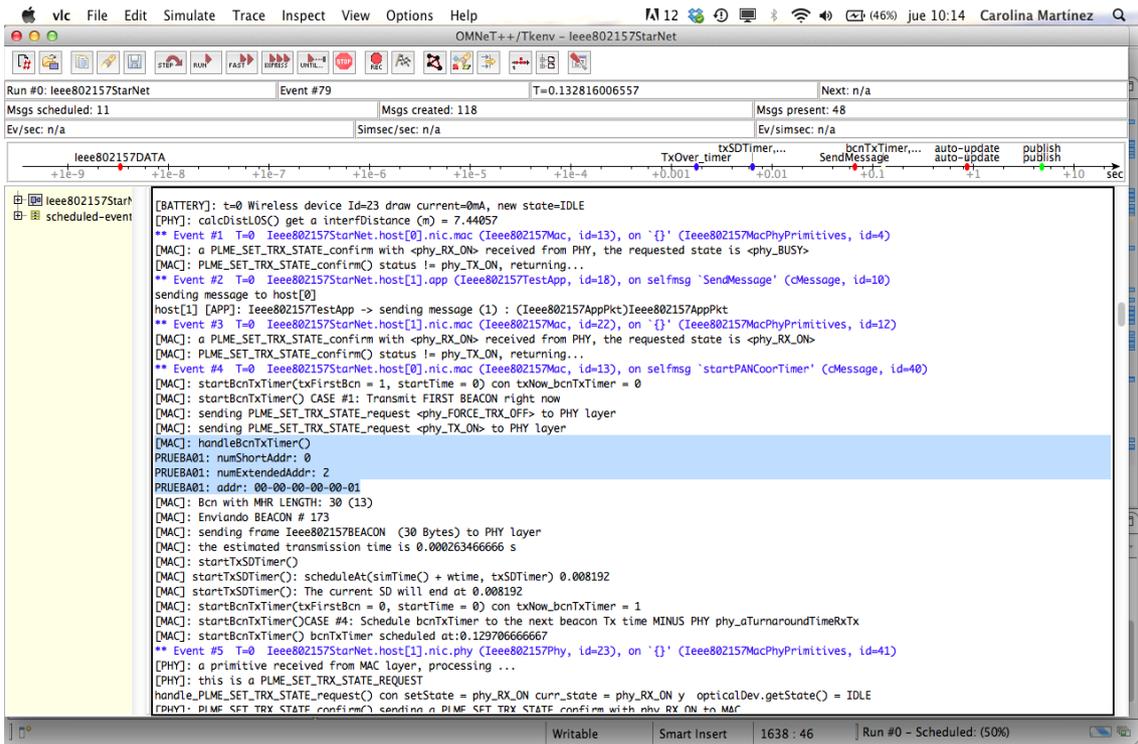


Figura 6.11. Configuración de una prueba del modo indirecto iniciada por coordinador.

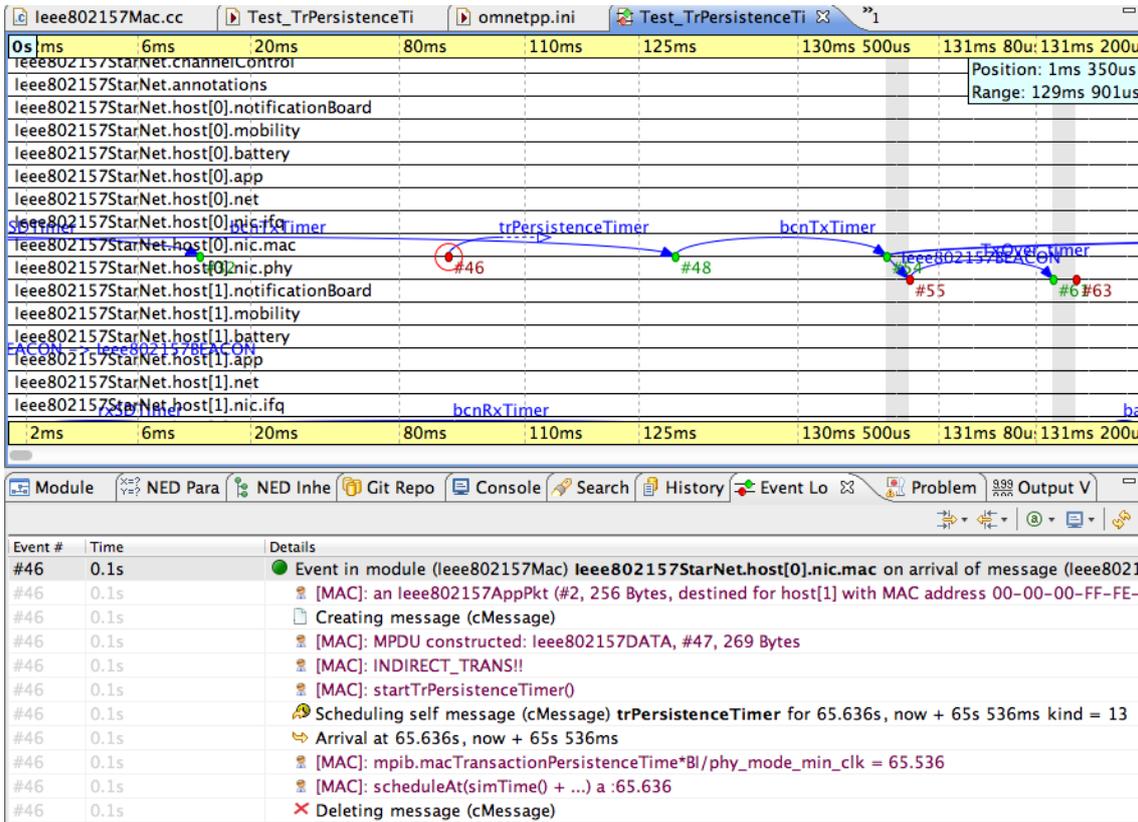


Figura 6.12. Resultado de la simulación del modo indirecto iniciada por coordinador.

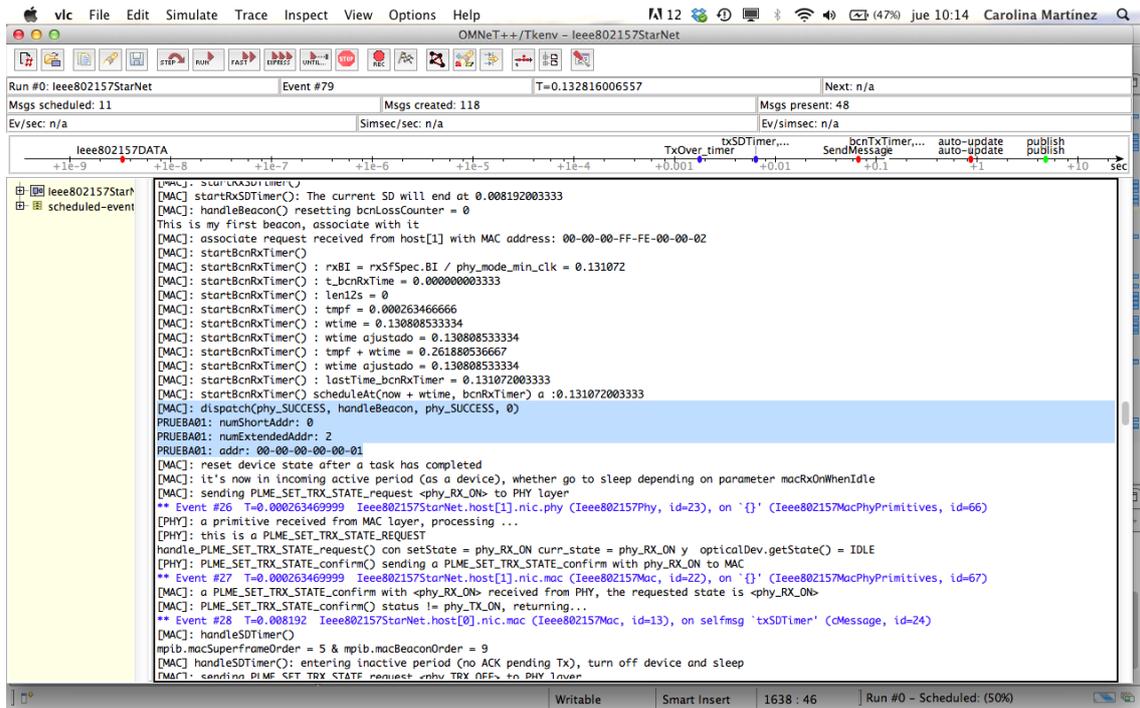


Figura 6.13. Configuración de una prueba del modo indirecto iniciada por coordinador: caso lista llena.

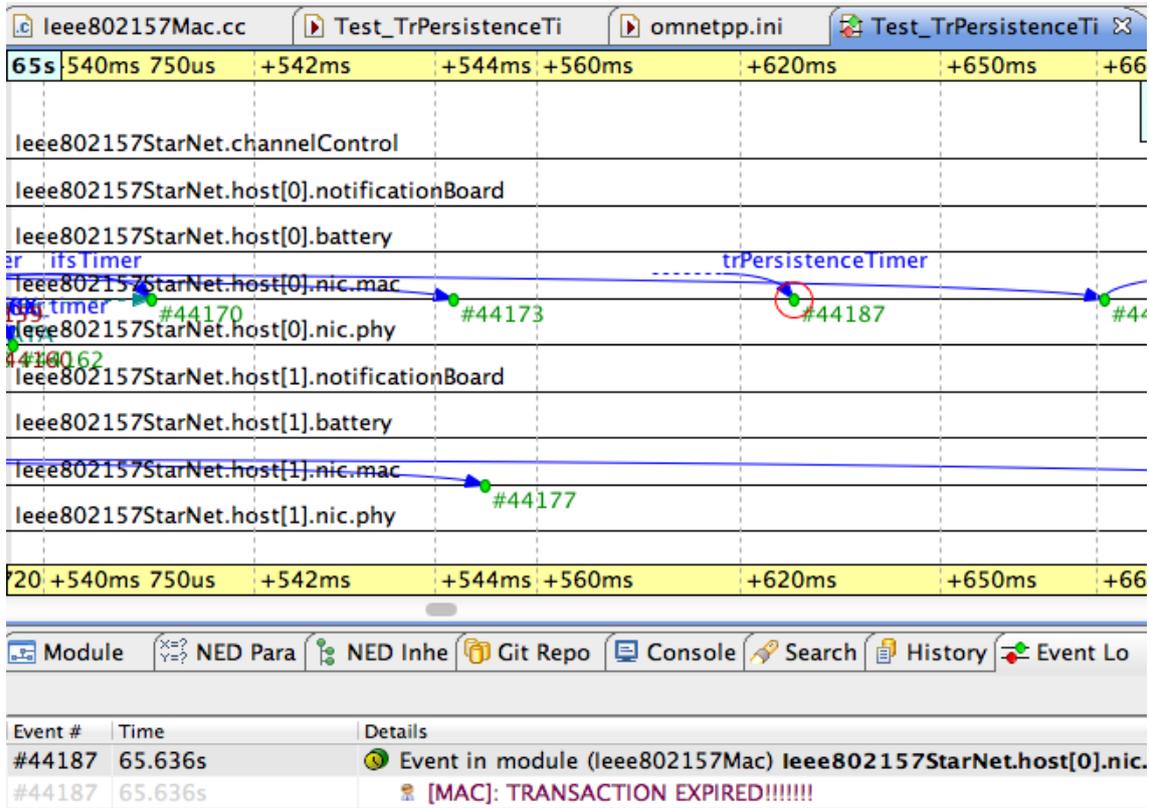


Figura 6.14. Resultado de la simulación del modo indirecto iniciada por coordinador: caso lista llena.

## 6.7. Conclusiones

En este capítulo se describe la parametrizan las simulaciones. Se simulan escenarios y condiciones concretas para comprobar el funcionamiento de los mecanismos implementados de sistemas VLC, y más concretamente, de las características de la capa MAC. El capítulo recoge la configuración de esas simulaciones, su ejecución, tratamiento de los problemas encontrados y análisis de los resultados obtenidos.



# Capítulo 7

## Conclusiones y Líneas futuras

---

*"La felicidad es saber unir el final con el principio."*

Pitágoras (569a.C., 475a.C.)

### 7.1. Objetivos alcanzados

El objetivo principal de este proyecto fue seguir la línea marcada por dos proyectos anteriores, donde se abordaba la implementación de la capa física, a implementar el nivel de capa de acceso al medio, protocolo MAC del estándar IEEE 802.15.7, con la finalidad de modelar redes VLC en la herramienta de simulación OMNeT++, y posibilitar su estudio mediante la simulación, comportamiento frente a nodos ocultos, tasas de tráfico, retardos, etc.

En esta memoria se ha descrito el desarrollo de un paquete software que implementa el nivel MAC de la norma IEEE802.15.7 para incluirla como un módulo dentro de la herramienta OMNet. El módulo MAC desarrollado permite modelar las principales funciones, protocolos y servicios de nivel MAC para comunicaciones ópticas no guiadas de corto alcance usando luz visible, en base a lo establecido en el estándar IEEE-802.15.7. El nuevo módulo del simulador se ha creado bajo el entorno OMNET++.

El objetivo principal de este proyecto final de carrera fue crear una versión del protocolo MAC que soportase los siguientes puntos:

1. Modo con baliza habilitada.
2. Mecanismo de acceso múltiple por detección de portadora con y sin evasión de colisiones (CSMA y CSMA/CA) para mecanismo de acceso con baliza ranurado.
3. Modelo de transferencia directo y GTS (ranuras de tiempo garantizadas).
4. Control de espaciado entre tramas, filtrado y detección de duplicidad.
5. Control de temporizaciones según la unidad de tiempo de relojes ópticos.
6. Implementación de las primitivas necesarias para el funcionamiento de los puntos anteriores.
7. Creación del formato de tramas para VLC, incluyendo las tramas CVD para mantener la intensidad de luz apropiada entre tramas de datos, soportar control de luminosidad, y para el suministro de información visual tales como el estado de la comunicación y la calidad del canal al usuario

En la comunicación directiva VLC el campo de visión es limitado, por lo que los mecanismos especificados en el estándar pueden ser insuficientes para un completo control de evasión de colisiones, es el llamado problema del nodo oculto. Por ese motivo, este proyecto final de carrera también pretende ser soporte para simulación de distintas propuestas de mecanismos que solventen dicho problema.

Aunque existe la posibilidad de configurar la red con las topologías de punto a punto, estrella o difusión, el desarrollo se limitó al caso de redes de topología en estrella. Además, se hizo uso del marco de trabajo existente en el proyecto INETMANET sobre el protocolo para WPAN de baja velocidad, definido en el estándar IEEE.802.15.4, como base para el desarrollo del código.

Aunque el propósito principal es establecer un modelo de la capa MAC e integrarlo con el módulo de la capa física, de forma que resulte plenamente funcional y se puedan ejecutar simulaciones y obtener resultados. Para ello, fue necesario modificar elementos de otras capas, introducir parametrizaciones, incluir nuevos módulos o modificar los existentes.

Esto ha permitido diseñar variedad de simulaciones con escenarios y condiciones concretas para estudiar, analizar y comprobar el funcionamiento de la capa MAC y los mecanismos de contienda de acceso de las redes de comunicaciones ópticas de luz visible.

También se hizo necesario modificar el método de configuración de las simulaciones basado en una serie de parámetros de entrada a los que se pueden asignar valores. La parametrización establecida permite configurar diferentes escenarios y condiciones a través de un amplio conjunto de parámetros que el usuario pueda establecer con facilidad de cara a sus simulaciones. Esta filosofía de interfaz de configuración parametrizable entronca con la propia del entorno de simulaciones OMNET++ sobre el cual se ha trabajado.

Finalmente se ha logrado completar el sistema de recogida de los resultados de simulación. En este aspecto el entorno de simulaciones OMNET++ empleado ofrece gran cantidad de herramientas que ayudan a la recolección de resultados, elaboración de estadísticas y posterior organización de datos y representación gráfica de los mismos.

Las simulaciones ejecutadas, y centradas en verificar características y funciones de la capa MAC, permiten concluir que la implementación de tal capa es correcta y acorde a las especificaciones descritas en la norma IEEE-802.15.7.

Como conclusión general, podemos decir que el uso de VLC es una solución más de las comunicaciones en redes de área personal (PAN). Los sistemas VLC presentan gran versatilidad ya que se proporcionan diferentes aplicaciones tradicionales, como la iluminación de estancias, carteles luminosos, etc., junto con el uso de las comunicaciones. Tal y como se ha visto en este trabajo, el uso de estas comunicaciones presenta unos requerimientos para que la transmisión de información se realice de forma efectiva.

También hay que destacar que VLC no remplazará a las tecnologías inalámbricas actuales como WiFi o IrC, sino que se complementarán, ya que los sistemas VLC necesitan visibilidad directa y es posible que la transmisión no llegue a todos los lugares deseados, además complementando ambas tecnologías, se podrán solucionar las desventajas individuales de cada una de ellas.

### **7.2. Líneas de trabajo futuras**

A la vista del trabajo descrito en esta memoria, podemos concluir que se considera cumplido el objetivo principal propuesto en este PFC de desarrollar un paquete software para simulaciones de redes VLC que modela las principales funciones, protocolos y servicios de nivel MAC. Sin embargo, como se ha comentado anteriormente, la

funcionalidad introducida en el nuevo módulo sólo soporta la configuración de red en estrella.

Por tanto, la línea futura de trabajo más inmediata consistiría en ampliar las topologías de red soportadas. La modularidad de la arquitectura de los módulos desarrollados garantiza la posibilidad de incluir la implementación de nuevos módulos y protocolos de las capas altas (transporte y de aplicación), asegurándose la optimización gracias a la arquitectura modular y orientada a objetos del modelo de redes VLC implementado.

Asimismo, por idénticas razones de modularidad, cualquier mejora futura dictada por la actualización del estándar, podrá ser fácilmente integrable o modificada dentro del simulador.



# 8

## Pliego de Condiciones

---

### 8.1. Objeto

El presente pliego tiene por objeto establecer las condiciones generales para la implementación de las librerías necesarias del protocolo MAC del estándar IEEE802.15.7 resultante de la realización de este proyecto. Igualmente determina las condiciones técnicas específicas, referidas tanto a equipamiento hardware como a software requerido, que permiten el uso del dicho paquete de simulaciones. También se incluyen especificaciones a considerar para la correcta instalación de los módulos desarrollados dentro de la plataforma OMNeT++ sobre la cual es operativo y del paquete de simulaciones INETMANET desde el cual se aprovechan recursos.

Con todo ello, se pretende asegurar el funcionamiento fiable del paquete de simulaciones creado en este proyecto, de forma que se puedan realizar simulaciones de diferentes escenarios con seguridad de que los resultados de las mismas sean fiables. Es responsabilidad del usuario del simulador su correcta utilización (configuración de escenarios, interpretación de resultados, etc...).

### 8.2. Condiciones generales

El paquete software de simulaciones *simVLC* para la plataforma OMNeT++ generado en este proyecto, incluidos todos sus módulos componentes, tiene la consideración de software libre, con las implicaciones que ello conlleva. El propietario de OMNeT++ del mismo tiene libertad para su uso, copia, distribución y modificación. Se ofrece el código fuente del mismo. Si bien es posible realizar modificaciones sobre los componentes del simulador (clases, tipos de módulos y mensajes, etc.), será responsabilidad del usuario verificar el correcto funcionamiento tras las mismas.

Los módulos desarrollados están diseñados para trabajar bajo el entorno de simulaciones OMNeT++ versión 4.2.2. Además requiere de mantenerse referenciado al paquete de simulaciones INETMANET versión 2.0, pues hace uso de recursos del mismo. Por ello es necesario que *simVLC* e INETMANET coexistan en un espacio de trabajo del IDE de OMNeT++ a fin de poderse ejecutar simulaciones. Sólo se asegura el correcto funcionamiento del simulador *simVLC* con las versiones de la plataforma de simulaciones OMNeT++ y del paquete INETMANET testadas en este PFC.

### 8.3. Condiciones específicas

Dado el carácter software del proyecto, no existen requisitos técnicos más allá de las impuestos por los elementos hardware y las aplicaciones o plataformas software empleadas en el desarrollo del simulador *simVLC*. Por tanto, las especificaciones se dividen en equipamiento informático y en software requerido.

### 8.3.1. Especificaciones de equipamiento informático

La instalación del simulador *simVLC* se debe realizar sobre un equipo informático con unas características mínimas que aseguren una ejecución fluida de las simulaciones.

- Procesador Intel Dual Core (32 bits), o equivalentes.
- Memoria RAM con capacidad de al menos 2GBytes.
- Espacio de almacenamiento en disco de al menos 2GBytes (para OMNeT++, INETMANET y *simVLC*).
- Espacio extra de almacenamiento para ficheros de resultados, en caso de simulaciones con una alta carga de métricas de salida.

La portabilidad de los paquetes software involucrados en este PFC permite la utilización de plataformas PC y/o MAC.

### 8.3.2. Especificaciones de equipamiento informático

Se incluyen aquí el sistema operativo y las herramientas software necesarias para la correcta ejecución del simulador *simVLC*.

- Sistema operativo Windows XP (32bits) o superior. También se puede instalar sobre sistemas Linux o MAC.
- Plataforma de simulaciones OMNeT++, versión 4.2.2.
- Paquete de simulaciones INETMANET versión 2.0, incluyendo el modelo IEEE-802.15.4 para redes WSN.
- Paquete de simulaciones *simVLC*.
- Compilador de C++ (normalmente incorporado con el paquete OMNET++).

### 8.3.3. Instalación del paquete *simVLC*

Por tanto, la línea futura de trabajo más inmediata consistiría en ampliar las topologías de red soportadas. La modularidad de la arquitectura simulador *simVLC* garantiza la posibilidad de incluir la implementación de nuevos módulos, asegurándose la optimización gracias a la arquitectura modular y orientada a objetos del modelo de redes VLC implementado.

Para poder realizar simulaciones con el modelo de redes VLC incluido en *simVLC* se debe incorporar tal paquete de simulaciones al IDE de la plataforma OMNeT++. También se debe incorporar el paquete de simulaciones INETMANET 2.0, pues incorpora recursos de los que *simVLC* hace uso. Recordar que el paquete de simulaciones *simVLC* debe mantenerse referenciado a INETMANET.

Los paquetes de simulaciones, una vez incorporados en el IDE, reciben el nombre de proyectos. Para incorporar ambos paquetes de simulaciones o proyectos a un espacio de trabajo del IDE de OMNeT++ se deben seguir los siguientes pasos, los cuales suponen importar un proyecto ya existente (ver **Figura 8.1**):

1. Desde el IDE de OMNeT++, ir al menú *File->Import*.
2. En el cuadro de diálogo que nos sale escoger *General ->Existing Project into wokrspace*.

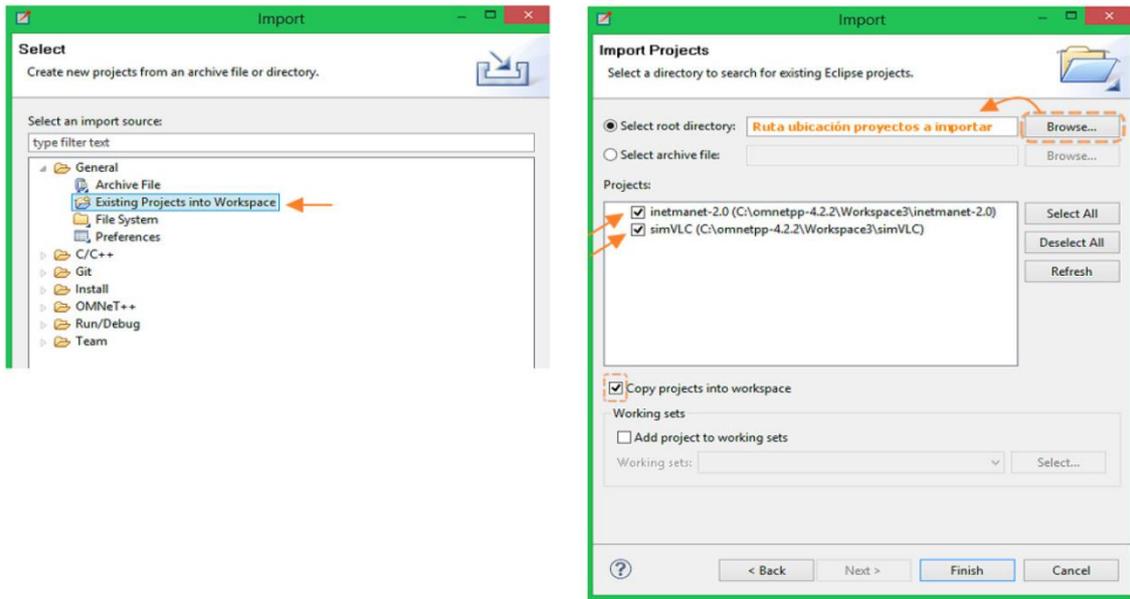


Figura 8.1. Cuadros de diálogos para la importación de proyectos existentes a un espacio de trabajo de OMNeT++.

3. Buscar la carpeta albergando ambos paquetes (en nuestro caso en el CD del proyecto).
4. Marcar ambos, indicando que se copien dentro del espacio de trabajo.
5. Importar.

Tras esto ya tendremos ambos proyectos en el espacio de trabajo activo del IDE de OMNeT++. Podremos hacer multitud de operaciones, como añadir nuevos elementos, configurar nuevas simulaciones, modificar el código de clases, recompilar, inspeccionar elementos, y lo más importante, ejecutar simulaciones.

Los proyectos importados, ya estaban previamente compilados, y por tanto se pueden ejecutar simulaciones directamente. Si se realizará alguna modificación del código C++, formato de mensajes o tipos de módulos se debe recompilar el proyecto para que los cambios tengan efecto en las simulaciones

### 8.3.4. Uso básico del simulador *simVLC*

Una vez instalado el simulador *simVLC* en la plataforma de simulaciones OMNeT++, y con presencia del paquete de simulaciones INETMANET referenciado, un usuario normal (no programador) ya puede ejecutar simulaciones del modelo de redes VLC interiores de corto alcance en línea de visión. Existen dos opciones de uso:

- a. Ejecutar escenarios existentes, ya configurados a priori.
- b. Configurar nuevos escenarios a partir de los existentes, reconfigurando los valores de los parámetros de entrada en los ficheros *.ini*.

La primera opción implica no adaptar el escenario de red VLC a simular a ningún caso concreto pretendido por el usuario. Permite visualizar las simulaciones creadas como parte de este proyecto y que modelan interesantes escenarios preestablecidos. Todos los escenarios preestablecidos se ejecutan a partir de una única configuración de lanzador creada.

Para ello se selecciona el proyecto *simVLC* en el explorador de OMNeT++, y con éste seleccionado se pulsa sobre el botón "*Iniciar simulación*" (ver Figura 8.2). Se nos preguntará, de entre una lista de escenarios preestablecidos, cual se desea ejecutar. Seguidamente comenzará la ejecución de la simulación del escenario escogido.

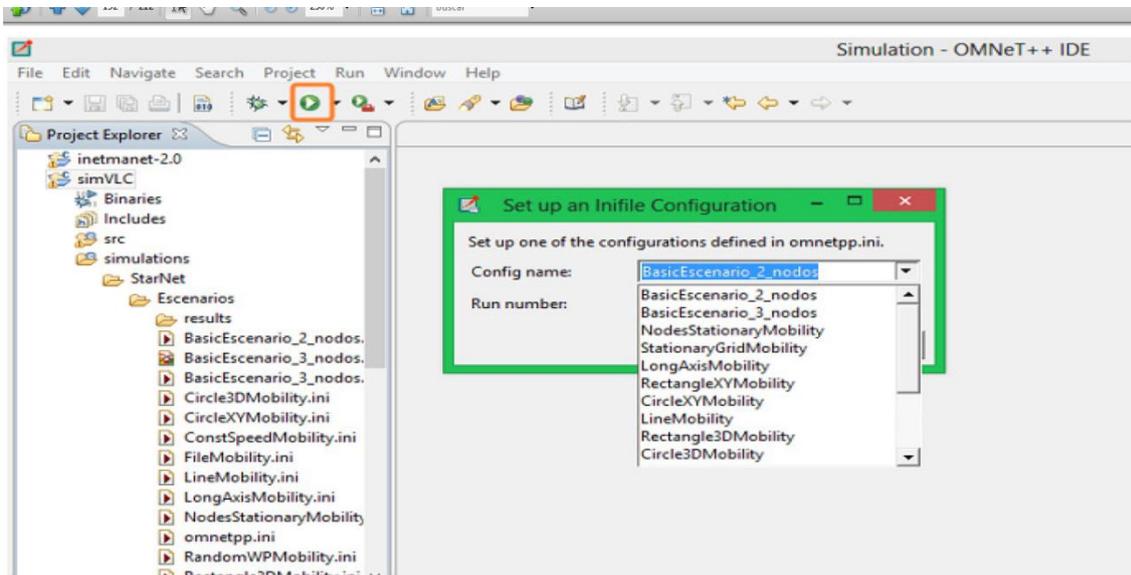


Figura 8.2. Botón de inicio de simulaciones, con diferentes escenarios configurados en el lanzador.

Se podrá detener la simulación cuando se desee con el botón "*Parar simulación*". Para más información sobre las opciones del lanzador de las simulaciones y del control de las mismas, consultar los manuales de OMNeT++ [6]

La segunda opción para ejecutar la simulación de un escenario es la realmente útil para cualquier usuario. Consiste en configurar los valores de los parámetros NED que definen el escenario pretendido. Para poder configurar un escenario concreto es necesario conocer todos los parámetros del modelo de simulaciones *Ieee802157StarNet* definidos a lo largo de todos los módulos y su significado. Estos parámetros reciben valores en los diferentes archivos *.ini* existentes en el simulador. Para dar valores a los parámetros NED del modelo existen ciertas reglas que se pueden consultar en los manuales de OMNeT++ [6].

De forma simplificada se puede buscar el parámetro pretendido en los ficheros *.ini*, ya sea en el general *omnetpp.ini* o en las secciones definidas en otros archivos *.ini* y asignarles el valor adecuado para las condiciones de simulación pretendidas. Tras esto se lanza la simulación como se indicó para la primera opción, con el botón "Iniciar simulación" y escogiendo el escenario adecuado. En la Figura 8.3 se observa la forma de asignar valores a los parámetros, utilizando la nomenclatura de nombres de módulos NED del modelo, finalizados con el nombre del parámetro.

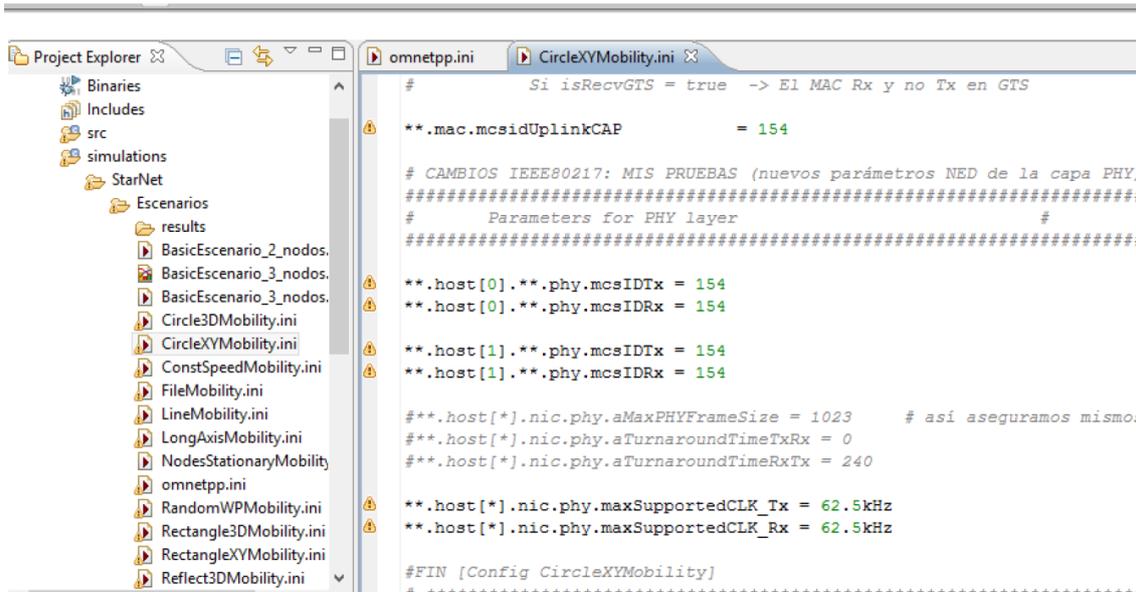


Figura 8.3. Captura de fichero de configuración de parámetros NED abierto.

Una vez terminada una simulación, se podrá acceder a los resultados de la misma a través de los ficheros de resultados creados en la carpeta `/simVLC/simulations/results`. Se remite a los manuales de OMNeT++ para las fases de estudio de los resultados [6].



# 9

## Presupuesto

---

### 9.1. Objeto

El presente presupuesto aporta un cálculo del coste total del desarrollo de los módulos que implementan el protocolo de la capa MAC del estándar IEEE 802.15.7. El total resultante básicamente se subdivide en costes de personal, derivados de la dedicación del Ingeniero de Telecomunicaciones ejecutor del paquete software, y los costes de recursos materiales, derivados de la amortización de los recursos de tipo hardware y software empleados durante el desarrollo.

### 9.2. Coste de personal

El cálculo del subtotal por este concepto se basa en la estimación de la temporización dedicada a cada una de las fases atravesadas durante el desarrollo del paquete de simulaciones realizado en este PFC. Tal estimación temporal se muestra en la siguiente **Tabla 9.1**.

**Tabla 9.1.** Aproximación de la dedicación temporal en las distintas fases del desarrollo de la librería MAC del IEEE 802.15.7.

FASE	TAREA	DEDICACIÓN (semanas)
ESTUDIOS PREVIOS	Documentación inicial y estudio del entorno	1
	Estudio de los estándares (especificaciones)	2
	Familiarización con herramientas software	2
ANÁLISIS (especificaciones)	Análisis de la implementación del modelo MAC	8
DISEÑO E IMPLEMENTACIÓN	Diseño e implementación del nuevo modelo de simulaciones	16
DEPURACIÓN Y EVALUACIÓN	Depuración y evaluación de resultados	3
DOCUMENTACIÓN	Generación de memoria descriptiva y otros documentos	10

Se deduce una dedicación temporal de 44 semanas en total, con una media de trabajo de 20 horas semanales (media jornada). Así pues, en horas, la dedicación temporal total estimada para la ejecución de las diferentes fases del proyecto es de 880 horas.

Se deduce una dedicación temporal de 44 semanas en total, con una media de trabajo de 20 horas semanales (media jornada). Así pues, en horas, la dedicación temporal total estimada para la ejecución de las diferentes fases del proyecto es de 880 horas.

El Colegio Oficial de Ingenieros de Telecomunicaciones (COIT) [84] ha dejado de publicar los baremos orientativos por honorarios para proyectos de telecomunicaciones, promovido por una nota informativa emitida por el Ministerio de Economía y Hacienda, en la que se recuerda que siguiendo directrices europeas se deben eliminar tales baremos orientativos de honorarios. En esa nota se indica que dichos honorarios son libres y responden al libre acuerdo entre el profesional y el cliente.

Aun así, en este proyecto se ha estimado el coste de personal tomando como referencia los precios por hora recogidos en los últimos baremos publicados. Ellos establecen que para la realización de un proyecto de telecomunicaciones de este tipo, el coste por hora sea de 58 euros, con un coeficiente de reducción del 0,45 para trabajos que superen las 720 horas pero no alcancen las 1080 horas, como es el caso. Por tanto, el coste total por personal se representa en la siguiente **Tabla 9.2**.

**Tabla 9.2.** Costes por dedicación temporal de personal, o mano de obra.

TOTAL DE HORAS	COSTE/HORA (€)	COEFIC. DE REDUCCIÓN (720 < horas < 1080)	TOTAL COSTES POR PERSONAL (€)
880	58	0.45	28.072

### 9.3. Coste de recursos materiales

Los recursos materiales empleados en la realización de este proyecto software se pueden agrupar en tres tipos. Igualmente se clasifican los costes derivados de ellos.

- Costes Hardware:** Incluyen la amortización de los equipos informáticos empleados.
- Costes Software:** Incluyen la amortización de los paquetes de software comercial que han sido necesarios. Gran parte de las herramientas empleadas son software libre, por lo que no suponen coste alguno.
- Otros costes:** Incluyen otros gastos genéricos como la conexión ADSL, el material de oficina, los costes en reprografía, etc.

El desglose de estos costes se presenta en la siguiente **Tabla 9.3**, de la cual se obtiene el subtotal en este concepto.

**Tabla 9.3.** Desglose de los costes por recursos materiales.

Recurso	Precio (€)	Tasa Anual de Amortización	Período de Uso (meses)	SUBTOTAL (€)
Ordenador Portátil (IBM PC)	700	0.5	11	320,9
Otro hardware (almacenamiento masivo, impresoras, etc...)	80	0.5	11	36,7
Software comercial (sistema operativo Win8 y Office 2007)	290	0.3	11	79,75
Software libre (OMNET++, INETMANET, Doxygen, etc...)	0	---	11	0
Otros recursos (ADSL, electricidad, consumibles, etc..)	400	0.1	11	36,7
Gastos de encuadernación	120	1	---	120
<b>TOTAL COSTES POR RECURSOS (€) =</b>				<b>594,05</b>

#### 9.4. Coste del proyecto

El coste total por la realización del presente proyecto de telecomunicaciones, teniendo en cuenta el 7% de impuesto general indirecto canario (IGIC), asciende a **30.672,67 €**.



# 10

## Bibliografía

---

- [1] Javier Areitio Bertolín, “Identificación y análisis de la tecnología VLC/Li-Fi desde la perspectiva de su ciberseguridad-ciberprivacidad”, Conectónica, Abril, 2013.
- [2] Bernard Llanos Flores, “Visible Light Communications”, Proyecto Fin de Carrera, Universidad Politécnica de Catalunya, 2014.
- [3] “Visible Light Communication Consortium”, <http://www.vlcc.net>, 2013.
- [4] “Japan Electronics and Information Technology Industries Association”, <http://www.jeita.or.jp>, 2015.
- [5] IEEE Standard for Local and Metropolitan Area Networks - Part 15.7 “Short-Range Wireless Optical Communication Using Visible Light”, Sep 2011. Available: <http://standards.ieee.org/getieee802/download/802.15.7-2011.pdf>
- [6] OMNeT++ Discrete Event Simulator, <http://www.omnetpp.org>.
- [7] INET Framework. <https://inet.omnetpp.org/>
- [8] IEEE 802.15.4 Model for OMNeT++/INET Framework. <http://www.ccsllabs.org/~dressler/projects/zigbee/802154>.
- [9] Roberto H. Medina Sosa, “Implementación software del nivel físico (PHY) del estándar IEEE-802.15.7 (VLC) para la plataforma de simulaciones OMNET++”, Proyecto Fin de Carrera, Universidad de Las Palmas de Gran Canaria, 2015.
- [10] Carlos Ley Bosch, “Análisis de los efectos del problema del nodo oculto en las redes IEEE 802.15.7. Propuesta de solución”, Tesis Doctoral, Universidad de Las Palmas de Gran Canaria, 2015.
- [11] N. Sklavos, M. Hübner, D. Goehringer, P. Kitsos. *System-Level Design Methodologies for Telecommunications*, Springer, 2013, Capítulo 1, “History of Visible Communications”.
- [12] “Wikipedia fotófono”. [Disponible online]: <https://en.wikipedia.org/wiki/Photophone>.
- [13] “Wikipedia Flash-Matic Tuning”. [Disponible online]: [https://en.wikipedia.org/wiki/Zenith\\_Flashmatic](https://en.wikipedia.org/wiki/Zenith_Flashmatic).

- [14] Bernard Llanos Flores, “Visible Light Communications”, Proyecto Fin de Carrera, Universidad Politécnica de Catalunya, 2014.
- [15] Major J. D. Harris, “Wire at war - Signals communication in the South African War”, 1899-1902.
- [16] Bruce, Robert Vance, “Bell: Alexander Bell and the Conquest of Solitude”, p. 336. Ithaca, New York: Cornell University Press (1990).
- [17] Groth, Mike, “Photophones Revisted”, *Amateur Radio Magazine*, April 1987 pp. 12–17 and May 1987 pp. 13–17.
- [18] Blanca Lorenzo Grandes, “Estudio del Estado del Arte de los sistemas de comunicaciones por luz visible (VLC)”, Proyecto Fin de Carrera, Universidad de Sevilla, 2015.
- [19] Blog *Visible Light Communications*. [Disponible online]: <http://visiblelightcomm.com/what-is-visible-light-communication-vlc>.
- [20] N. Sklavos, M. Hübner, D. Goehringer, P. Kitsos. *System-Level Design Methodologies for Telecommunications*, Springer, 2013, Capítulo 1, “Introduction”.
- [21] “Wikipedia VLC”. [Disponible online]: [https://en.wikipedia.org/wiki/Visible\\_light\\_communication](https://en.wikipedia.org/wiki/Visible_light_communication).
- [22] “Wikipedia RONJA”. [Disponible online]: <https://en.wikipedia.org/wiki/RONJA>.
- [23] C. Pohlmann, “Visible Light Communications”, [Disponible online]: <http://www.slideshare.net/hossamzein/visible-light-communication>, 2010.
- [24] Ke Wang, Ampalavanapillai Nirmalathas, Christina Lim, Efstratios Skafidas, “High-speed indoor optical wireless communication system with single channel imaging receiver”, *Optics Express*, vol. 20, no. 23, pp. 25356-25356, 2012.
- [25] G. Cossu, A. M. Khalid, P. Choudhury, R. Corsini, E. Ciaramella, “3.4 Gbit/s visible optical wireless transmission based on RGB LED”, *Optics Express*, vol. 20, no. 26, pp. B501-B506, 2012.
- [26] A. M. Khalid, G. Cossu, R. Corsini, P. Choudhury, E. Ciaramella, “1-Gb/s transmission over a phosphorescent white LED by using Rate-Adaptive Discrete Multitone Modulation”, *IEEE Photonics Journal*, vol. 4, no. 5, pp. 1465 - 1473, Oct. 2012.

- [27] H. L. Minh, D. O'Brien, G. Faulkner, L. Zeng, K. Lee, D. Jung, Y. Oh, "High-speed visible light communications using multiple-resonant equalization", *IEEE Photonics Technology Letters*, vol. 20, no.14, pp. 1243-1245, July 2008.
- [28] Lubin Zeng, Dominic C. O'Brien, Hoa Le Minh, Grahame E. Faulkner, Kyungwoo Lee, Daekwang Jung, YunJe Oh, Eun Tae Won, "High data rate multiple input multiple output (MIMO) optical wireless communications using white LED lighting", *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 9, December 2009.
- [29] D. Tsonev, S. Sinanovic, H. Haas, "Practical MIMO Capacity for Indoor Optical Wireless Communication with White LEDs". *IEEE 77th Vehicular Technology Conference (VTC Spring)*, 2013.
- [30] Giulio Cossu, Amir M. Khalid, Raffaele Corsini, Ernesto Ciaramella, "Non-Directed Line-of-Sight Visible Light System providing high-speed and robustness to ambient light". *Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC)*, 2013.
- [31] J. Vucic, C. Kottke, A. Buttner, K. D. Langer, J. W. Walewski, "White-light wireless transmission at 200+ Mbit/s net data rate by use of discrete-multitone modulation", *IEEE Photon. Technol. Lett.*, vol. 21, no. 20, pp. 1511–1513, 2009.
- [32] Jelena Grubor, Sian Chong Jeffrey Lee, Klaus-Dieter Langer, Ton Koonen, Joachim W. Walewski, "Wireless high-speed data transmission with phosphorescent white-light LEDs". *33rd European Conference and Exhibition of Optical Communication*, 2007.
- [33] Sayed Ehsan Alavi, Hoda Rezaie, Abu Sahmah M. Supa'at, "Integrated system of Visible Free Space Optic with PLC". *IEEE Asia-Pacific Conference on Applied Electromagnetics (APACE)*, 2010.
- [34] A.R. Ndjiongue, H.C. Ferreira, K. Ouahada, A.J.H. Vinckz, "Low-complexity SOCPBFSK-OOK interface between PLC and VLC channels for low data rate transmission applications". *18th IEEE International Symposium on Power Line Communications and its Applications (ISPLC)*, 2014.
- [35] T. Komine, N. Nakagawa, "Integrated system of white led visible light communication and power line communication", *IEEE Transactions on Consumer Electronics*, vol. 49, no. 1, pp. 71-79, Feb. 2003.

- [36] Cree Inc., “Cree reaches LED Industry Milestone with 200 Lumen-Per-Watt LED”. Press release, 2012. [Disponible online]: <http://www.cree.com/News-and-Events/Cree-News/Press-Releases/2012/December/MKR-intro>. December 18, 2012.
- [37] R. Perez-Jimenez, J. Rufo, C. Quintana, J. Rabadan, F.J. Lopez-Hernandez, “Visible Light Communication Systems for Passenger In-Flight Data Networking”. *IEEE International Conference on Consumer Electronics (ICCE)*, 2011.
- [38] Stanley R. Mohler, J. Robert Dille, Harry L. Gibbons. *Circadian rhythms and the effects of long distance flights*, US Dept. of Transportation, Federal Aviation Administration, Office of Aviation Medicine, 1968. [Disponible online]: <http://catalogue.nla.gov.au/Record/4185738>.
- [39] Kang Tae Gyu *et al.*, “IEEE 802.15.7. Regulation Document Configuration. Spectrum Frequency”, IEEE P802.15 Working Group for Wireless Personal Area Networks”, Samsung Electronics Co. LTD, University of Oxford, ETRI, VLCC. 2008.
- [40] Peter K. Kaiser. *The joy of virtual perception*, web book, 1996. [Disponible online]: <http://www.yorku.ca/eye>.
- [41] Kang Tae Gyu *et al.*, “IEEE 802.15.7. Regulation Document Configuration. Product Safety”, IEEE P802.15 Working Group for Wireless Personal Area Networks”, Samsung Electronics Co. LTD, University of Oxford, ETRI, VLCC. 2008.
- [42] Ajmme Kajros, “Harald Haas: datos inalámbricos en cada foco incandescente”, TED, Agosto 2011. [Disponible online]: [https://www.ted.com/talks/harald\\_haas\\_wireless\\_data\\_from\\_every\\_light\\_bulb?language=es](https://www.ted.com/talks/harald_haas_wireless_data_from_every_light_bulb?language=es).
- [43] J. E. De Ángeles Barriga, “Estudio y simulación de un sistema ACO-OFDM para comunicaciones ópticas inalámbricas”, Universidad de Sevilla, 2014, Capítulo 3.
- [44] Nishant Bharti, Nishant Gupta, R. Kritika, Saurabh Singh, “Visible Light Communication”, BTech project, University of Delhi, 2012, Capítulo 1 “Visible Light Communication”. [Disponible online]: [https://xa.yimg.com/kq/groups/.../name/NISHANT\\_BHARTI\\_NSIT.pdf](https://xa.yimg.com/kq/groups/.../name/NISHANT_BHARTI_NSIT.pdf)

- [45] Nishant Bharti, Nishant Gupta, R Kritika, Saurabh Singh, “Visible Light Communication”, BTech project, University of Delhi, 2012, Capítulo 3 “Merits and Advantages”. [Disponible online]: [https://xa.yimg.com/kq/groups/.../name/NISHANT\\_BHARTI\\_NSIT.pdf](https://xa.yimg.com/kq/groups/.../name/NISHANT_BHARTI_NSIT.pdf).
- [46] N. Sklavos, M. Hübner, D. Goehringer, P. Kitsos. *System-Level Design Methodologies for Telecommunications*, Springer, 2013, Capítulo 3, “VLC: Advantages and Disadvantages”.
- [47] Swook Hann, Jung-Hun Kim, Soo-Yong Jung, Chang-Soo Park, “White LED Ceiling Lights Positioning Systems for Optical Wireless Indoor Applications”. *36th European Conference and Exhibition on Optical Communication (ECOC)*, 2010.
- [48] Soo-Yong Jung, Swook Hann, Chang-Soo Park. “TDOA-based optical wireless indoor localization using LED ceiling lamps”, *IEEE Transactions on Consumer Electronics*, vol. 57, no. 4, pp. 1592–1597, November 2011.
- [49] S. H. Yang, E. M. Jeong, D. R. Kim, H. S. Kim, Y. H. Son, S. K. Han, “Indoor three-dimensional location estimation based on LED visible light communication”, *Electronics Letters*, vol. 49, no. 1, pp. 54-56, January 2013.
- [50] Youngsuk Kim, Junho Hwang, Jisoo Lee, Myungsik Yoo, “Position estimation algorithm based on tracking of received light intensity for indoor visible light communication systems”. *Third International Conference on Ubiquitous and Future Networks (ICUFN)*, 2011.
- [51] M.S. Rahman, M.M. Haque, Ki-Doo Kim, “High precision indoor positioning using lighting LED and image sensor”. *14th International Conference on Computer and Information Technology (ICCIT)*, 2011.
- [52] Heather Brundage, “Designing a wireless underwater optical communication system”. Massachusetts Institute of Technology, 2010. [Disponible online]: <https://dspace.mit.edu/handle/1721.1/57699>.
- [53] Felix Schill, Uwe R. Zimmer, Jochen Trunpf, “Visible Spectrum Optical Communication and Distance Sensing for Underwater”. *Proceedings of Australian Conference on Robotics and Automation (ACRA)*, 2004.
- [54] Cen Bi Liu, “The Case for Vehicular Visible Light Communication (V2LC): Architecture, Services and Experiments”, M.S. Thesis, Rice University, 2010. [Disponible online]: <http://gradworks.umi.com/14/86/1486556.html>.

- [55] Dewayne Washington, “Laser Demonstration Reveals Bright Future for Space Communication”. NASA Press release, December 2013. [Disponible online]: <https://www.nasa.gov/content/goddard/laser-demonstration-reveals-bright-future-for-space-communication>
- [56] Elizabeth Landau, “OPALS: Light beams let data rates soar”. NASA Press release, December 2014. [Disponible online]: [https://www.nasa.gov/mission\\_pages/station/research/news/opals\\_data\\_rates\\_soar](https://www.nasa.gov/mission_pages/station/research/news/opals_data_rates_soar)
- [57] M. Miki, E. Asayama, T. Hiroyasu, “Intelligent lighting system using visible-light communication technology”. *IEEE Conference on Cybernetics and Intelligent Systems*, 2006.
- [58] Philips, “Meet Hue”. [Disponible online]: <http://www.meethue.com/en-US>.
- [59] Niovi Pavlidou, A. J. Han Vinck, Javad Yazdani, Bahram Honary, “Power line communications: State of the art and future trends”, *IEEE Communications Magazine*, vol. 41, no. 4, pp. 34 – 40, 2003.
- [60] M. Kavehra, F. Amirshahi, “Hybrid MV-LV power lines and white light emitting diodes for triple-play broadband access communications”. *IEEE Consumer Communications and Networking Conference*, 2006.
- [61] Alexander Vavoulas, Harilaos G. Sandalidis, Theodoros A. Tsiftsis, and Nicholas Vaiopoulos, “Coverage Aspects of Indoor VLC Networks”, *Journal of Lightwave Technology*, vol. 33, no. 23, pp. 4915-4921, 2015.
- [62] Dominic C. O’Brien. *Cooperation in Optical Wireless Communications*, Springer, 2007. Capítulo 32 “Cognitive Wireless Networks”, pp. 623-634.
- [63] J. Hou, D. C. O’Brien, “Vertical handover-decision-making algorithm using fuzzy logic for the integrated Radio-and-OW system”, *IEEE Transactions on Wireless Communications*, vol. 5, no. 1, pp. 176-185, Jan. 2006.
- [64] C. Quintana, V. Guerra, J. Rufo, J. Rabadan, R. Perez-Jimenez, “Reading lamp-based visible light communication system for in-flight entertainment”, *IEEE Transactions on Consumer Electronics*, vol. 59, no. 1, pp. 31-37, February 2013.
- [65] Y. F. Liu, C. H. Yeh, C. W. Chow, Y. Liu, Y. L. Liu, H. K. Tsang, “Demonstration of bi-directional LED visible light communication using TDD traffic with mitigation of reflection interference”, *Optics Express*, vol. 20, no. 21, pp. 23019-23024, 2012.
- [66] Yuanquan Wang, Yiguang Wang, Nan Chi, Jianjun Yu, Huiliang Shang, “Demonstration of 575-Mb/s downlink and 225-Mb/s uplink bi-directional

- SCM-WDM visible light communication using RGB LED and phosphor-based LED”, *Optics Express*, vol. 21, no. 1, pp. 1203-1208, 2013.
- [67] Etai Rosenkratz, Shlomi Arnon, “Modulated retro-reflector for VLC applications”. Proc. *SPIE 9224, Laser Communication and Propagation through the Atmosphere and Oceans III*, October 2014.
- [68] T. Borogovac, M. B. Rahaim, M. Tuganbayeva, T. Little, “Lights-off Visible Light Communications”. *2011 IEEE Globecom Workshops (GC’11)*, 2011.
- [69] “Visible Light Communications Association”, [Disponibile online]: <http://vlca.net/about-vlca>.
- [70] “Telecommunications Technology Association”; <http://www.tta.or.kr>, 2015.
- [71] “OMEGA, the Home Gigabit Access project”; <http://www.ict-omega.eu>, 2011.
- [72] “Institute of Electrical and Electronic Engineers”; <http://www.ieee.org>, 2015.
- [73] “Li-Fi Consortium”; <http://www.lificonsortium.org>, 2015.
- [74] “IEEE 802.15 WPANTM 15.7 Revision: Short-Range Optical Wireless Communications Task Group (TG 7r1)”; [http://www.ieee802.org/15/pub/IEEE\\_802\\_15\\_WPAN\\_15\\_7\\_Revision1\\_Task\\_Group.htm](http://www.ieee802.org/15/pub/IEEE_802_15_WPAN_15_7_Revision1_Task_Group.htm), 2015.
- [75] “IEEE Standard for Information Technology – Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks -Specific Requirements - Part 2: Logical Link Control”; ISO 8802-2 IEEE 802.2, First Edition 1989-12-31 (Revision of IEEE Std 802.2-1985), pp. 1 - 114, 2013.
- [76] “IEEE Standard for Local and Metropolitan Area Networks, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)”. New York. Septiembre 2006.
- [77] S. McCanne and S. Floyd. Network Simulator. <http://www.isi.edu/nsnam/ns/>
- [78] OPNET Technologies, Inc., Opnet Modeler Wireless Suite - ver. 11.5A, <http://www.opnet.com>
- [79] GNS3. <https://gns3.com/>
- [80] CNET Network Simulator. <http://www.csse.uwa.edu.au/cnet/>
- [81] J-Sim. <http://www.j-sim.org/>
- [82] Philip Baldwin, Sanjeev Kohli, Edward A. Lee, Xiaojun Liu, and Yang Zhao, " VisualSense: Visual Modeling for Wireless and Sensor Network Systems," Technical Memorandum UCB/ERL M05/25, University of California, Berkeley, CA 94720, USA, July 15, 2005.
- [83] INETMANET, <https://github.com/aarizaq/inetmanet-2.0>
- [84] Colegio Oficial de Ingenieros de Telecomunicación, <http://www.coit.es>



# Anexo I

## Primitivas de servicio de la capa MAC

### 0. Introducción

La capa MAC proporciona una interfaz entre el SSCS (subcapa de convergencia específica de servicio), DME (entidad de gestión del dispositivo) y la capa física PHY. La capa MAC conceptualmente incluye una entidad de gestión llamada MLME (*medium-access-control link-management entity*, entidad de gestión del enlace en el control de acceso al medio).

Esta entidad proporciona las interfaces de servicio a través de las cuales las funciones de gestión pueden ser invocadas. El MLME también es responsable de mantener una base de datos de objetos gestionados pertenecientes a la subcapa MAC. Esta base de datos es conocida como PIB (capa física personal del área de información de la base de la red) de la capa MAC.

La **Figura I.1** representa los componentes e interfaces de la capa MAC. La capa MAC proporciona los siguientes dos servicios, a los que se accede a través de dos puntos de acceso SAP (service access point):

- El **servicio de datos MAC**, se accede a través de la subcapa de parte común MAC (MCPS) del SAP de datos (MCPS-SAP).
- **Servicio de gestión MAC**, se accede a través del MLME-SAP.

Estos dos servicios proporcionan la interfaz entre el SSCS y la capa PHY, a través de las interfaces PD-SAP y PLME-SAP. Además, para esas interfaces externas, también existe una interfaz implícita entre la MLME y el MCPS que permite que la MLME use el servicio de datos MAC.

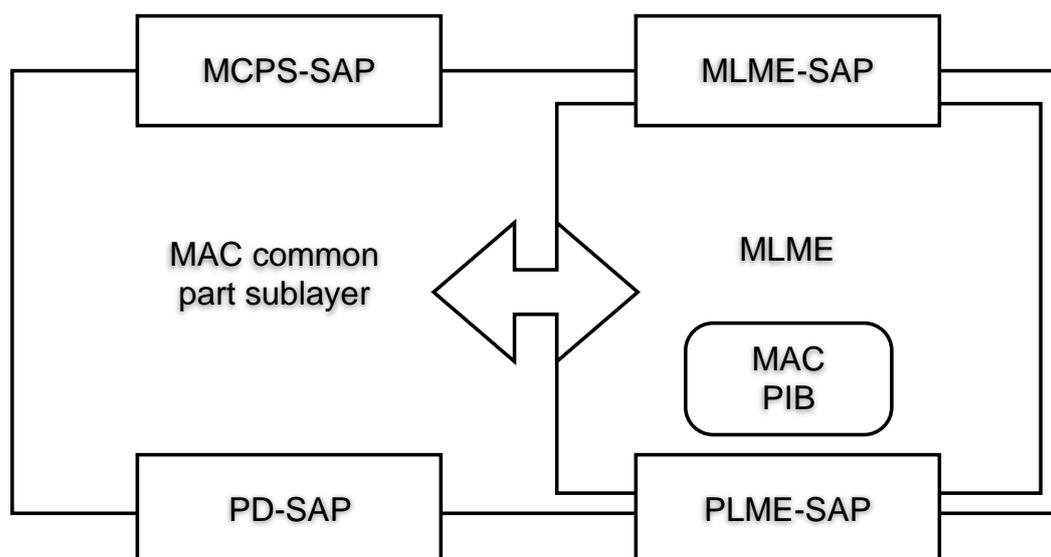


Figura I.1. Modelo de referencia para la subcapa MAC.

Tabla I.1. Primitivas MCPS-SAP

Primitiva MCPS-SAP	Request	Confirm	Indication
MCPS-DATA	?	?	?
MCPS-PURGE	?	?	--

## I.1. Servicio de datos MAC

El MCPS-SAP soporta el transporte de las unidades de dato del protocolo SSCS (SPDUs) entre entidades SSCS pares. En la **Tabla I.1** se resumen las primitivas del servicio de datos MAC.

### I.1.1. MCPS-DATA.request

Solicita la transferencia de un dato SPDU desde una entidad local SSCS a una entidad pareja SSCS única. En el modo empaquetado, múltiples MSDU se pasan mediante una entidad local SSCS a una entidad pareja SSCS única.

Soporta los siguientes parámetros:

- ***DstAddrMode***: (tipo entero, rango válido 0x00 – 0x03) indica el modo de dirección destino para esta primitiva y MPDU subsiguiente. Este valor puede tomar los siguientes valores:
  - 0x00 = Sin dirección
  - 0x01 = Sin campo de dirección (broadcast)
  - 0x02 = Dirección corta de 16-bits
  - 0x03 = Dirección extendida de 64-bits
- ***DstVPANId***: (tipo entero, rango válido 0x0000 – 0xffff) indica el identificador de VPAN de 16-bits de la entidad a la cual la MSDU está siendo transferido.
- ***DstAddr***: (tipo dirección de dispositivo, rango válido según especifica *DstAddrMode*) indica la dirección de dispositivo individual de la entidad a la cual la MSDU está siendo transferido.
- ***MsdLength***: (tipo entero, rango válido  $\leq aMaxMACPayloadSize$ ) indica el número de octetos contenidos en la MSDU para ser transmitida mediante la entidad de la subcapa MAC.
- ***Msd***: (tipo conjunto de octetos, no requiere validez de rango) indica el número de octetos forman el MSDU para ser transmitido por la entidad de la subcapa MAC.
- ***MsdHandle***: (tipo entero, rango válido 0x00 – 0xff) indica el control asociado con el MSDU para ser transmitido por la entidad de la subcapa MAC.
- ***TxOptions***: (tipo bitmap, campo de 3 bits) el campo de tres bits ( b0, b1, b2) indica las opciones de transmisión para este MSDU:
  - b<sub>0</sub> = 1 → transmisión confirmada
  - b<sub>0</sub> = 0 → transmisión no confirmada
  - b<sub>1</sub> = 1 → transmisión GTS para una VPAN con baliza
  - b<sub>1</sub> = 0 → transmisión CAP para una VPAN con baliza (sin baliza siempre

es 0)

$b_2 = 1 \rightarrow$  transmisión indirecta

$b_2 = 0 \rightarrow$  transmisión directa

- **SecurityLevel:** (tipo entero, rango válido 0x00 – 0x07) indica el nivel de seguridad que es usado.
- **KeyIdMode:** (tipo entero, rango válido 0x00 – 0x03) indica el modo usado para identificar la clave que es usada. Este parámetro es ignorado si el parámetro *SecurityLevel* es ignorado o establecido a 0x00.
- **KeySource:** (tipo 0, 4, u 8 octetos, rango según especifica el *KeyIdMode*) indica la fuente originaria de la clave a ser usada. Este parámetro es ignorado si el parámetro *KeyIdMode* es ignorado o establecido a 0x00.
- **KeyIndex:** (tipo entero, rango válido 0x01 – 0xff) indica el índice de la clave a ser usada. Este parámetro es ignorado si el parámetro *KeyIdMode* es ignorado o establecido a 0x00.
- **DataRate:** (tipo enumeración, campo de 6 bits) indica la tasa de datos de la trama PHY para ser transmitido por la entidad PHY.
- **BurstMode:** (tipo booleano, rango válido TRUE or FALSE) se establecerá al valor TRUE si el modo ráfaga está siendo usado, si no a FALSE.
- **ColorReceived:** (tipo booleano, rango válido TRUE or FALSE) se establecerá al valor TRUE si se envía la trama de confirmación y es usada la función de color para la indicación de estado de confirmación en la trama CVD.  
Se establecerá al valor FALSE si se envía la trama de confirmación pero no es usada la función de color para la indicación de estado de confirmación en la trama CVD.
- **ColorNotReceived:** (tipo booleano, rango válido TRUE or FALSE) se establecerá al valor TRUE si se envía la trama de confirmación y es usada la función de color para la indicación de estado de confirmación en la trama CVD.  
Se establecerá al valor FALSE si se envía la trama de confirmación pero no es usada la función de color para la indicación de estado de confirmación en la trama CVD.

### I.1.2. MCPS-DATA.confirm

Esta primitiva informa sobre el resultado de una petición de transferencia de datos SPDU (MSDU) de una entidad local SSCS a una entidad SSCS par. La primitiva MCPS-DATA.confirm puede devolver el estado SUCCESS, indicando que la petición de transmisión se ha realizado con éxito, o el código de error apropiado.

Soporta los siguientes parámetros:

- **MsdHandle:** (tipo entero, rango válido 0x00 – 0xff) indica El control asociado con el MSDU que está siendo confirmado.
- **status:** (tipo enumeración, rango válido) indica el estado del último MSDU transmitido. Puede tomar los siguientes valores:
  - SUCCESS,
  - TRANSACTION\_OVERFLOW,
  - TRANSACTION\_EXPIRED,
  - CHANNEL\_ACCESS\_FAILURE,
  - INVALID\_ADDRESS,

- INVALID\_GTS,
  - NO\_ACK,
  - COUNTER\_ERROR,
  - FRAME\_TOO\_LONG,
  - UNAVAILABLE\_KEY,
  - UNSUPPORTED\_SECURITY
  - or INVALID\_PARAMETER
- **Timestamp:** (tipo entero, rango válido 0x000000 – 0xffffffff) indica el tiempo, en relojes ópticos, al cual el dato fue transmitido. Es opcional.
- El valor de este parámetro se considerará válido solo si el valor del parámetro status es SUCCESS; si no es así el valor del parámetro *TimeStamp* no se usará para ningún otro propósito.
- El límite está descrito por el atributo *macTimeStampOffset*. El *TimeStamp* es un valor de 24-bits, y la precisión del valor será un mínimo de 20 bits, con los 4-bits más bajos comienza el menos significativo.

### I.1.3. MCPS-DATA.indication

Esta primitiva indica la transferencia de datos SPDU (MSDU) desde la subcapa MAC hasta la entidad SSCS local. En el modo empaquetado, se pasan múltiples MSDU a través de entidades SSCS locales a una entidad SSCS par.

Soporta los siguientes parámetros:

- **SrcAddrMode:** (tipo entero, rango válido 0x00 – 0x03) indica el modo de dirección fuente para esta primitiva y MPDU recibido. Este valor puede tomar los siguientes valores:
  - 0x00 = Sin dirección
  - 0x01 = Sin campo de dirección (broadcast)
  - 0x02 = Dirección corta de 16-bits
  - 0x03 = Dirección extendida de 64-bits
- **SrcVPANId:** (tipo entero, rango válido 0x0000 – 0xffff) indica el identificador de VPAN de 16-bits de la entidad desde la cual el MSDU fue recibido.
- **SrcAddr:** (tipo dirección de dispositivo, rango válido según especifica *SrcAddrMode*) indica la dirección de dispositivo individual de la entidad desde la cual el MSDU fue recibido.
- **DstAddrMode:** (tipo entero, rango válido 0x00 – 0x03) indica el modo de dirección destino para esta primitiva y MPDU recibido. Este valor puede tomar los siguientes valores:
  - 0x00 = Sin dirección
  - 0x01 = Sin campo de dirección (broadcast)
  - 0x02 = Dirección corta de 16-bits
  - 0x03 = Dirección extendida de 64-bits
- **DstVPANId:** (tipo entero, rango válido 0x0000 – 0xffff) indica el identificador de VPAN de 16-bits de la entidad a la cual la MSDU está siendo transferido.
- **DstAddr:** (tipo dirección de dispositivo, rango válido según especifica *DstAddrMode*) indica la dirección de dispositivo individual de la entidad a la cual la MSDU está siendo transferido.

- ***MsduLength***: (tipo entero, rango válido  $\leq aMaxMACPayloadSize$ ) indica el número de octetos contenidos en el MSDU que está siendo indicado mediante la entidad de la subcapa MAC.
- ***Msdu***: (tipo conjunto de octetos, no requiere validez de rango) indica el número de octetos forman el MSDU que está siendo indicado mediante la entidad de la subcapa MAC.
- ***MpduLinkQuality***: (tipo entero, rango válido 0x00 – 0xff) indica el valor WQI medido durante la recepción del MPDU. Valores bajos representan baja WQI.
- ***DSN***: (tipo entero, rango válido 0x00 – 0xff) indica el DSN de la trama de datos recibida.
- ***Timestamp***: (tipo entero, rango válido 0x000000 – 0xffffffff) indica el tiempo, en relojes ópticos, al cual el dato fue recibido. Es opcional.  
El límite está descrito por el atributo *macTimeStampOffset*. El *TimeStamp* es un valor de 24-bits, y la precisión del valor será un mínimo de 20 bits, con los 4-bits más bajos comienza el menos significativo.
- ***SecurityLevel***: (tipo entero, rango válido 0x00 – 0x07) indica nivel de seguridad supuestamente usado por la trama de datos recibida.
- ***KeyIdMode***: (tipo entero, rango válido 0x00 – 0x03) indica el modo usado para identificar la clave que supuestamente se usó por la fuente originaria de la trama recibida. Este parámetro es ignorado si el parámetro *SecurityLevel* es ignorado o establecido a 0x00.
- ***KeySource***: (tipo 0, 4, u 8 octetos, rango según especifica el *KeyIdMode*) indica la fuente originaria de la clave a ser usada. Este parámetro es ignorado si el parámetro *KeyIdMode* es ignorado o establecido a 0x00.
- ***KeyIndex***: (tipo entero, rango válido 0x01 – 0xff) indica el índice de la clave a ser usada. Este parámetro es ignorado si el parámetro *KeyIdMode* es ignorado o establecido a 0x00.
- ***DataRate***: (tipo enumeración, campo de 6 bits) indica la tasa de datos de la trama PHY para ser transmitido por la entidad PHY.
- ***BurstMode***: (tipo booleano, rango válido TRUE or FALSE) se establecerá al valor TRUE si el modo ráfaga está siendo usado, si no a FALSE.
- ***ColorReceived***: (tipo booleano, rango válido TRUE or FALSE) se establecerá al valor TRUE si la trama CVD es enviada cuando la trama de datos se recibe satisfactoriamente.
- ***ColorNotReceived***: (tipo booleano, rango válido TRUE or FALSE) se establecerá al valor TRUE si la trama CVD es enviada cuando la trama de datos no se recibe.

#### I.1.4. MCPS-PURGE.request

Esta primitiva es opcional para el dispositivo y permite a la capa superior eliminar una MSDU de la cola de transacción. Es opcional. Sólo admite el parámetro *MsduHandle*.

#### I.1.5. MCPS-PURGE.confirm

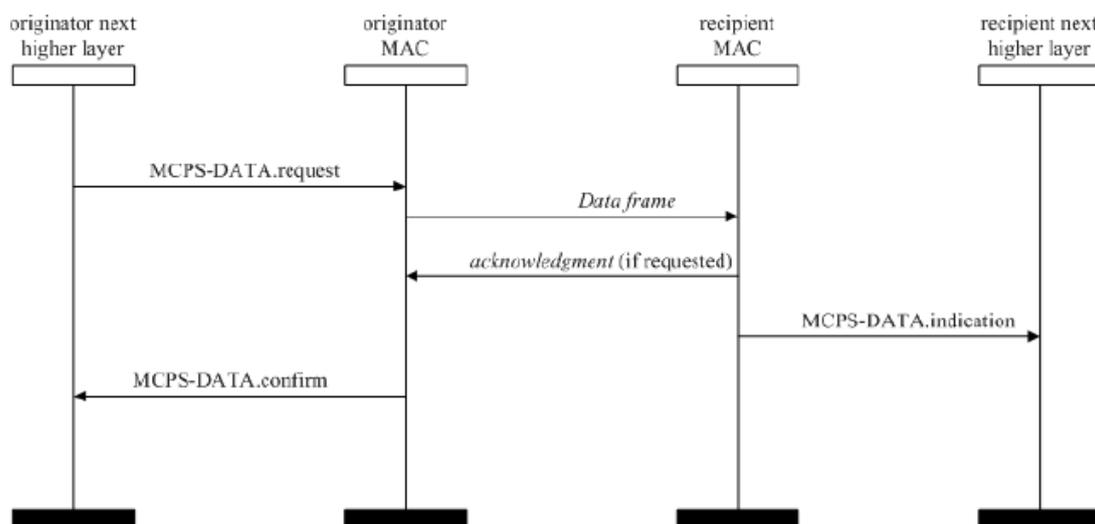


Figura I.2. Secuencia de mensajes describiendo el servicio de datos MAC.

Esta primitiva permite a la subcapa MAC notificar a la capa superior el éxito en su petición de eliminación de una MSDU de la cola de transacción. Se genera en la entidad de la subcapa MAC en respuesta a la primitiva MCPS-PURGE.request.

Esta primitiva devuelve el estado de SUCCESS, indicando que se ha realizado la eliminación con éxito, o INVALID\_HANDLE, indicando un error.

Admite los parámetros *MsduHandle* y *status*.

### I.1.6. Secuencia de mensajes del servicio de datos

En la Figura I.2 se ilustra la secuencia de mensajes para la correcta transferencia de datos entre dos dispositivos.

## I.2. Servicio de gestión MAC

El MLME-SAP permite el transporte de instrucciones de gestión entre la capa superior y el MLME. También define cómo un dispositivo debe asociarse con una VPAN. Todos los dispositivos deben proporcionar una interfaz para la petición y confirmación de las primitivas de asociación. Las primitivas de asociación de respuesta e indicación son opcionales en un dispositivo. Se clasifican en los siguientes tipos:

#### 1. Primitivas MLME-SAP (Tabla I.2 y I.3) con funciones:

- asociación.
- des-asociación.
- notificación de baliza.
- lectura de atributos PIB.
- gestión GTS.
- especificación del tiempo de habilitación del receptor.
- escaneo de canales.
- escritura de los atributos PIB.
- actualización de la configuración de supertrama.

- pérdida de sincronización con un coordinador.
2. Primitivas **PD-SAP** (Tabla I.4) para la petición de datos de un coordinador.
  3. Primitivas **PLME-SAP** (Tabla I.5) para el transporte de comandos.

Tabla I.2. Primitivas MLME-SAP

Primitiva MLME-SAP	Request	Confirm	Response	Indication
MLME-ASSOCIATION	?	?	?	?
MLME-DISSOCIATION	?	?	--	?
MLME-BEACON-NOTIFY	--	--	--	?
MLME-GET	?	?	--	--
MLME-GTS	?	?	--	?
MLME-RESET	?	?	--	--
MLME-RX-ENABLE	?	?	--	--
MLME-SCAN	?	?	--	--
MLME-COMM-STATUS	--	--	--	?

Tabla I.3. Primitivas MLME-SAP

Primitiva MLME-SAP	Request	Confirm	Response	Indication
MLME-SET	?	?	--	--
MLME-START	?	?	--	--
MLME-SYNC	?	--	--	--
MLME-SYNC-LOSS	--	--	--	?
MLME-POLL	?	?	--	--

Tabla I.4. Primitivas PD-SAP

Primitiva PD-SAP	Request	Confirm	Indication
PD-DATA	?	?	?

Tabla I.5. Primitivas PLME-SAP

Primitiva PLME-SAP	Request	Confirm
PLME-CCA	?	?
PLME-GET	?	?
PLME-SET	?	?
PLME-SET-TRX-STATE	?	?
PLME-SWITCH	?	?

## I.2.1. MLME-SAP: Primitivas de asociación

A continuación se describen el subconjunto de primitivas MLME-SAP dedicadas a la gestión de la asociación.

### I.2.1.1. MLME-ASSOCIATE.request

Esta primitiva permite al dispositivo solicitar la asociación con un coordinador. Es generada por la capa superior de un dispositivo no asociado y se emite a la MLME para realizar la solicitud. Si el dispositivo desea asociarse a través de un coordinador en una VPAN con uso de baliza, la MLME opcionalmente podrá realizar un seguimiento a la baliza del coordinador antes de la emisión de la primitiva.

### I.2.1.2. MLME-ASSOCIATE.indication

Esta primitiva se utiliza para indicar la recepción de una instrucción de petición de asociación. Se genera por parte de la MLME del coordinador y se emite a la capa superior para indicar la recepción de la instrucción.

### I.2.1.3. MLME-ASSOCIATE.response

Esta primitiva se utiliza para iniciar una respuesta a una primitiva MLME-ASSOCIATE.indication. Se genera por la capa superior del coordinador y se emite a su MLME para responder a la primitiva MLME-ASSOCIATE.indication.

### I.2.1.4. MLME-ASSOCIATE.confirm

Esta primitiva se utiliza para informar a la capa superior del dispositivo que inicia la petición sobre si la solicitud de asociación se ha realizado con éxito o no.

### I.2.1.5. Secuencia de mensajes de asociación

En la **Figura I.3** se muestra la secuencia de mensajes que se utilizan por un dispositivo que no realiza seguimiento de baliza al coordinador para asociarse a una VPAN.

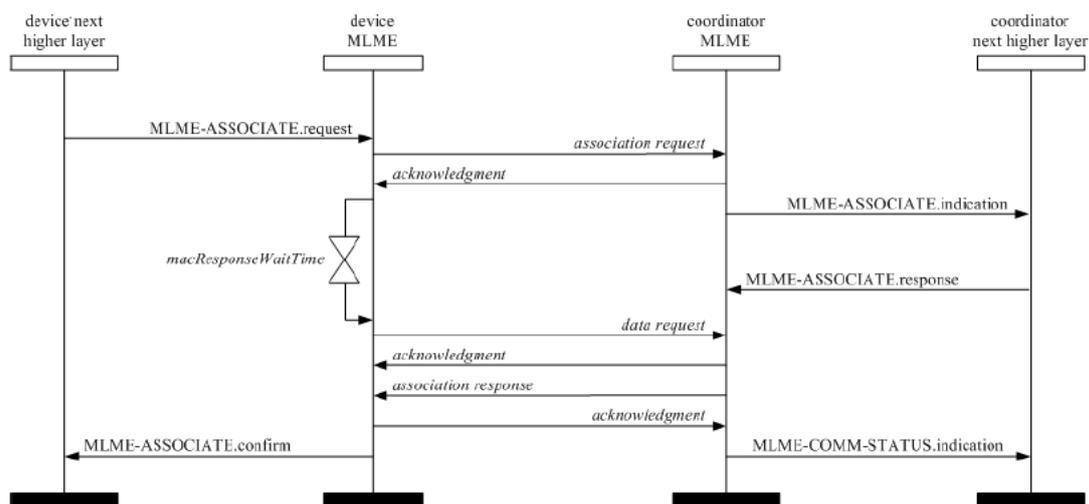


Figura I.3. Secuencia de mensajes para la asociación.

## I.2.2. MLME-SAP: Primitivas de des-asociación

Las primitivas de des-asociación MLME-SAP definen el proceso de des-asociación de un dispositivo de una VPAN. Todos los dispositivos proporcionarán una interfaz para estas primitivas.

### I.2.2.1. MLME-DISASSOCIATE.request

Esta primitiva se utiliza por los dispositivos asociados para notificar al coordinador que pretende salir de la VPAN. También se utiliza por el coordinador para indicar a los dispositivos asociados cómo dejar la VPAN. Se genera por la capa superior de un dispositivo asociado y es emitida por la MLME para solicitar la des-asociación de la VPAN.

### I.2.2.2 MLME-DISASSOCIATE.indication

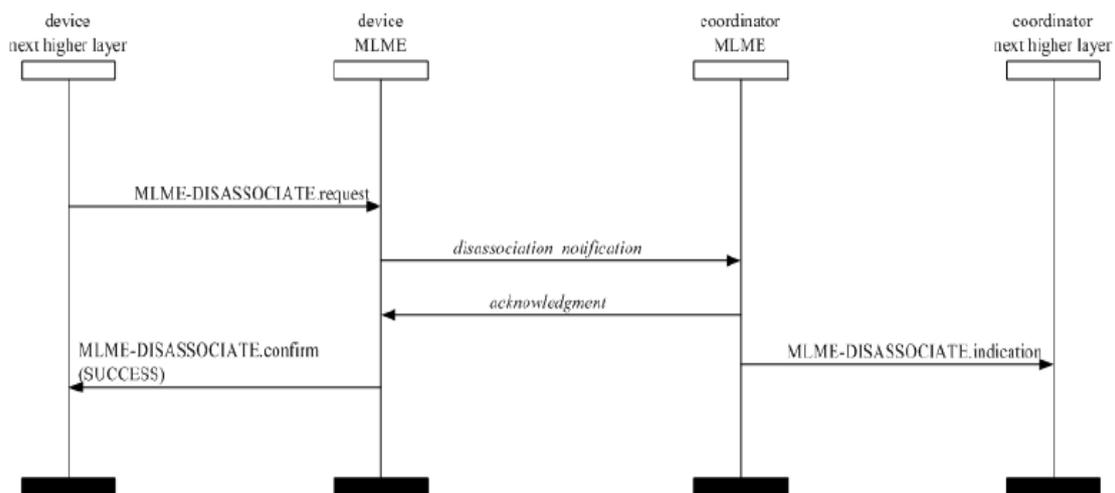
Esta primitiva se utiliza para indicar la recepción de una instrucción de notificación de des-asociación. Se genera por la MLME y se emite a la capa superior después de la recepción de la instrucción de notificación de des-asociación.

### I.2.2.3. MLME-DISASSOCIATE.confirm

Esta primitiva informa sobre el resultado de la primitiva MLME-DISASSOCIATE.request. Se genera por la MLME inicial y se emite a la capa superior en respuesta a la primitiva de petición. La primitiva responde el estado SUCCESS, indicando que la petición de asociación se ha realizado con éxito, o el código de error apropiado.

### I.2.2.4. Secuencia de mensajes de des-asociación

La petición de des-asociación puede originarse tanto por el dispositivo como por el coordinador. La **Figura I.4** ilustra la secuencia de mensajes necesaria para la des-asociación de un dispositivo de una VPAN.



**Figura I.4.** Secuencia de mensajes para la des-asociación iniciada por un dispositivo.

### **I.2.3. MLME-SAP: Primitiva de notificación de baliza**

La primitiva de notificación de baliza MLME-SAP define cómo se puede notificar a un dispositivo cuando la baliza se recibe durante las condiciones normales de operación. Todos los dispositivos proporcionarán una interfaz para las primitivas de notificación de baliza.

#### **I.2.3.1. MLME- BEACON-NOTIFY.indication**

Esta primitiva se utiliza para enviar a la capa superior los parámetros contenidos dentro de tramas de baliza recibidos por la subcapa MAC. La primitiva también envía una medida de la WQI y el tiempo en el que se ha recibido la trama de baliza.

### **I.2.4. MLME-SAP: Primitivas de lectura de atributos PIB**

Estas primitivas definen cómo leer los valores de los atributos PIB. Todos los dispositivos proporcionarán una interfaz para obtener esas primitivas.

#### **I.2.4.1. MLME-GET.request**

Esta primitiva solicita información sobre un atributo dado. Se genera por la capa superior y es emitida por su MLME para obtener información sobre el PIB.

#### **I.2.4.2 MLME-GET.confirm**

Esta primitiva informa sobre el resultado de una petición de información. Se genera por el MLME y se emite a la capa superior en respuesta a la primitiva MLME-GET.request. Esta primitiva devolverá el estado de SUCCESS, indicando que la respuesta a la lectura del atributo PIB se ha hecho de forma satisfactoria, o un código de error UNSUPPORTED\_ATTRIBUTE. Cuando se devuelva el código de error, el valor del parámetro PIBAttribute se establecerá a cero.

### **I.2.5. MLME-SAP: Primitivas de gestión GTS**

Las primitivas de gestión GTS MLME-SAP definen cómo se solicitan y gestionan los GTSs. Un dispositivo que desee utilizar estas primitivas o un GTS en general deberá realizar un seguimiento a las balizas de su coordinador. El uso de estas primitivas es opcional.

#### **I.2.5.1. MLME-GTS.request**

Esta primitiva permite al dispositivo enviar una petición al coordinador para asignar un nuevo GTS o rehusar uno existente. El coordinador también puede utilizarlas para iniciar la denegación de un GTS existente.

#### **I.2.5.2. MLME-GTS.indication**

Esta primitiva indica que el GTS ha sido asignado o que un GTS previamente asignado ha sido rehusado. Se genera por la MLME del coordinador a su capa superior siguiendo la recepción de una instrucción de petición de GTS.

#### **I.2.5.3. MLME-GTS.confirm**

Esta primitiva informa sobre el resultado de una petición de asignación de un nuevo GTS o rechazo de uno existente.

#### I.2.5.4. Secuencia de mensajes de gestión GTS

La gestión de asignación GTS puede originarse tanto por el dispositivo como por el coordinador. La **Figura I.5** ilustra la secuencia de mensajes necesaria para la asignación de un GTS iniciada por un dispositivo. La **Figura I.6** ilustra la secuencia de mensajes necesaria para el rechazo iniciado por un dispositivo (a) y el coordinador PAN (b).

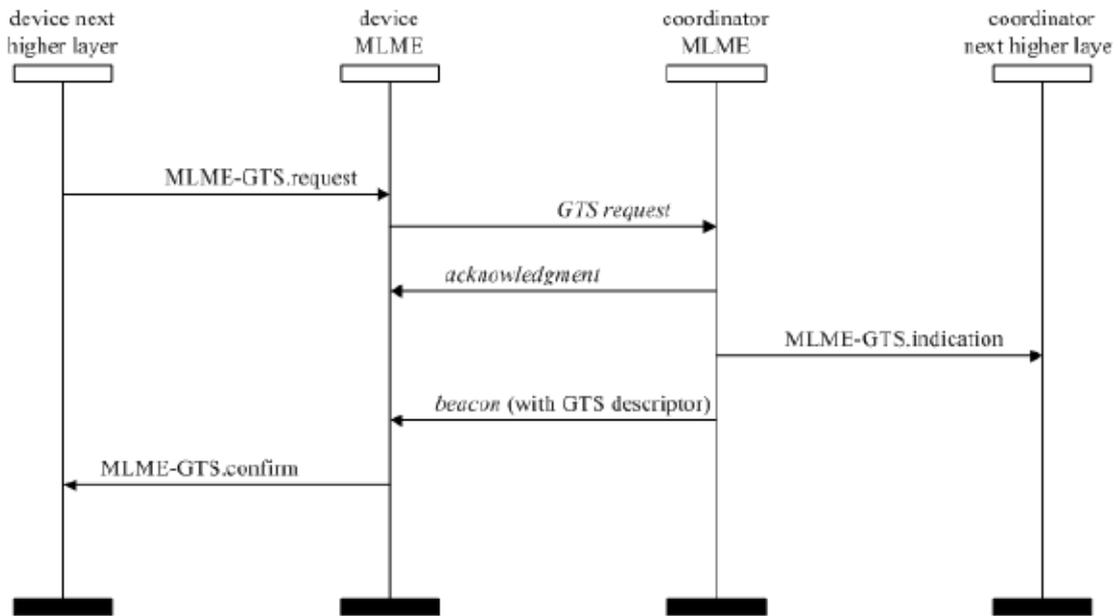


Figura I.5. Secuencia de mensajes para la asignación de un GTS iniciada por un dispositivo.

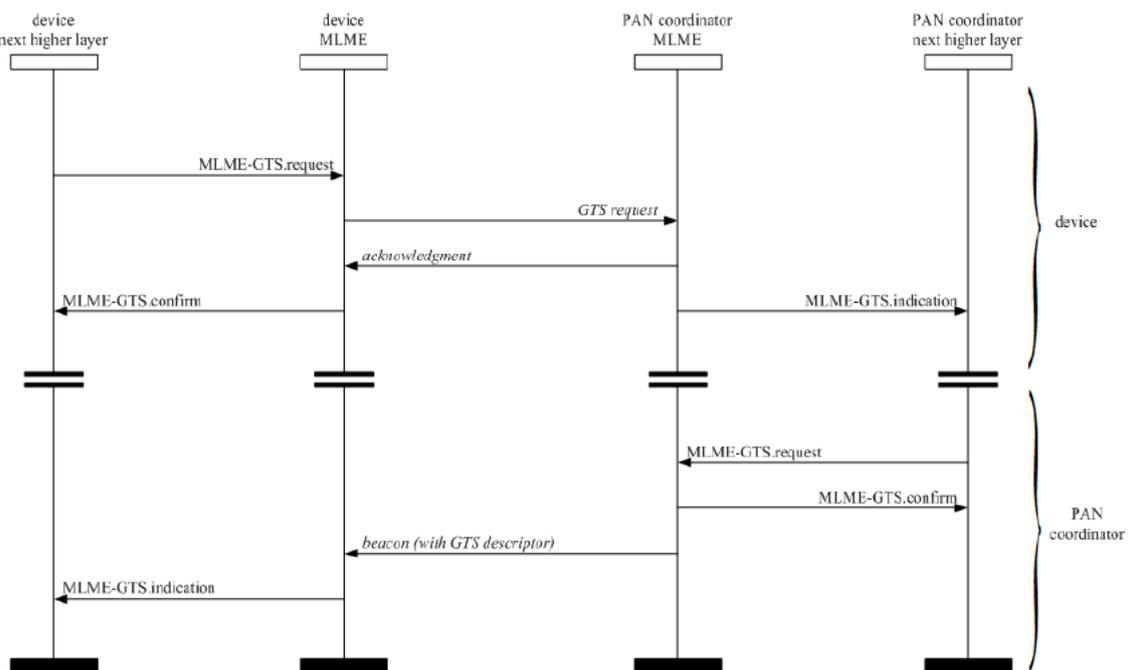


Figura I.6. Secuencia de mensajes para el rechazo iniciado por un dispositivo (a) y el coordinador PAN (b).

## **I.2.6. MLME-SAP: Primitivas para el reinicio de la subcapa MAC**

Las primitivas MLME-SAP para el reinicio especifican como reiniciar la subcapa MAC a los valores por defecto. Todos los dispositivos proporcionarán una interfaz para estas primitivas de reinicio.

### **I.2.6.1. MLME-RESET.request**

Estas primitivas permiten a la capa superior solicitar que la MLME realice una operación de reinicio. Se genera por la capa superior y se emite por la MLME para la petición de reinicio de la subcapa MAC a las condiciones iniciales. Esta primitiva se emite antes del uso de las primitivas MLME-START.request o MLME-ASSOCIATE.request.

### **I.2.6.2. MLME-RESET.confirm**

Esta primitiva informa sobre el resultado de la operación de reinicio. Se genera por la MLME y es emitida a la capa superior en respuesta a su primitiva de petición y siguiendo la recepción de la primitiva PLME-SET-TRX-STATE.confirm.

## **I.2.7. MLME-SAP: Primitivas para la especificación del tiempo de habilitación del receptor**

Las primitivas MLME-SAP de estado del receptor definen cómo un dispositivo puede habilitar o deshabilitar el receptor en un tiempo dado. Estas primitivas de estado de recepción son opcionales.

### **I.2.7.1. MLME-RX-ENABLE.request**

Esta primitiva permite a la capa superior solicitar que el receptor esté habilitado o deshabilitado un periodo de tiempo establecido. Se genera por la capa superior y es emitida por la MLME para habilitar el receptor en el instante de inicio de la supertrama actual o la siguiente en una VPAN con uso de baliza, o inmediatamente en una VPAN sin uso de baliza. Esta primitiva también se podrá generar para cancelar una petición de habilitación del receptor generada de forma previa. El receptor se habilitará o deshabilitará una vez por primitiva de petición.

### **I.2.7.2. MLME-RX-ENABLE.confirm**

Esta primitiva informa sobre el resultado del intento de habilitación o deshabilitación del receptor. Se genera por la MLME y se emite a la capa superior en respuesta a una primitiva MLME-RX-ENABLE.request. Devolverá el estado SUCCESS, si la petición de habilitación se realiza con éxito, o el código de error apropiado en caso contrario.

## **I.2.8. MLME-SAP: Primitivas para el escaneo de canales**

Las primitivas MLME-SAP de escaneo definen cómo puede determinar un dispositivo el uso de energía o la presencia o ausencia de VPAN en el canal de comunicación. Todos los dispositivos proporcionarán una interfaz para esas primitivas de escaneo.

### **I.2.8.1. MLME-SCAN.request**

Estas primitivas se utilizan para iniciar un escaneo de canal sobre una lista de canales. Un dispositivo puede utilizar un escaneo de canal para medir la energía del canal y buscar al coordinador con el cual está asociado, o buscar todos los coordinadores que estén transmitiendo tramas de baliza dentro del área de cobertura del mismo. Esta primitiva puede utilizarse para realizar un escaneo pasivo o activo con el objetivo de localizar tramas de baliza que contengan cualquier identificador VPAN.

### **I.2.8.2. MLME-SCAN.confirm**

Estas primitivas informan sobre el resultado de la petición de escaneo de canal. Se genera por la MLME y se emite a la capa superior cuando finaliza el escaneo de canal iniciado por la primitiva de petición. Devolverá el estado SUCCESS o el código de error apropiado.

### **I.2.7.2. MLME-RX-ENABLE.confirm**

Esta primitiva informa sobre el resultado del intento de habilitación o deshabilitación del receptor. Se genera por la MLME y se emite a la capa superior en respuesta a una primitiva MLME-RX-ENABLE.request. Devolverá el estado SUCCESS, si la petición de habilitación se realiza con éxito, o el código de error apropiado en caso contrario.

## **I.2.9. MLME-SAP: Primitivas del estado de comunicación**

Definen cómo se comunica la MLME con la capa superior para proporcionar información sobre el estado de transmisión, cuando la transmisión viene provocada por una primitiva de respuesta, y sobre los errores de seguridad en los paquetes entrantes. Todos los dispositivos proporcionarán una interfaz para la primitiva del estado de la comunicación.

### **I.2.9.1. MLME-COMM-STATUS.indication**

Esta primitiva permite a la MLME indicar el estado de la comunicación. Mediante la recepción de esta primitiva se notifica a la capa superior el estado de la comunicación y se notifica sobre cualquier error que haya ocurrido durante el procesamiento de seguridad de la trama entrante.

## **I.2.10. MLME-SAP: Primitivas para la escritura de los atributos PIB**

Las primitivas MLME-SAP de establecimiento definen cómo deben escribirse los atributos PIB.

### **I.2.10.1. MLME-SET.request**

Esta primitiva intenta la escritura de los valores dados a los atributos PIB que se indiquen.

### **I.2.10.2. MLME-SET.confirm**

Esta primitiva informa sobre el resultado del intento de escritura de un valor al atributo PIB.

## **I.2.11. MLME-SAP: Primitivas para la actualización de la configuración de supertrama**

La primitiva MLME-SAP *start* define cómo puede solicitar el coordinador una nueva configuración de supertrama con el objetivo de iniciar una VPAN o como comienza a transmitir balizas en una VPAN existente, para facilitar la detección de dispositivos o la parada de transmisión de balizas.

### **I.2.11.1. MLME-START.request**

Esta primitiva permite al coordinador VLC iniciar una nueva VPAN o el inicio de uso de una nueva configuración de supertrama. Esta primitiva también podrá utilizarse por un dispositivo que ya haya sido asociado a una VPAN existente para el inicio de una nueva configuración de supertrama.

### **I.2.11.2. MLME-START.confirm**

Esta primitiva informa sobre el resultado del inicio de uso de una nueva configuración de supertrama. Se genera en la MLME y se emite a la capa superior en respuesta a la primitiva de petición. Si la actualización de configuración se ha realizado de forma exitosa, el parámetro de estado indicará SUCCESS, de otra forma el parámetro indicará el error.

### **I.2.11.3. Secuencia de mensajes para la actualización de la configuración de supertrama**

La **Figura I.7** muestra la secuencia de mensajes necesaria para la actualización de supertrama para el caso de inicio transmisiones de baliza como coordinador.

## **I.2.12. MLME-SAP: Primitivas para la sincronización con un coordinador**

Las primitivas MLME-SAP de sincronización definen cómo se consigue la sincronización con un coordinador y cómo se comunica la pérdida de sincronización a la capa superior. Todos los dispositivos proporcionarán una interfaz para la primitiva de indicación. La primitiva de petición es opcional.

### **I.2.12.1. MLME-SYNC.request**

Esta primitiva solicita la sincronización con el coordinador mediante la adquisición, si está especificado, y el seguimiento de sus balizas. Se genera en la capa superior del dispositivo en una VPAN que permita el uso de baliza y se emite a su MLME para sincronizar con el coordinador.

## **I.2.13. MLME-SAP: Primitivas para la pérdida de sincronización con un coordinador**

### **I.2.13.1. MLME-SYNC-LOSS.indication**

Esta primitiva se genera por el MLME del dispositivo y se emite a la capa superior para indicar al coordinador la pérdida de sincronización. También se genera en la MLME del coordinador de VLC y se emite a la capa superior en caso de conflicto de identificador VPAN ID.

### I.2.13.2. Secuencia de mensajes para la sincronización con un coordinador

La Figura I.8 muestra la secuencia de mensajes necesaria para la sincronización del dispositivo con un coordinador. En la Figura I.8 (a) se emite la solicitud de sincronización única. En la Figura I.8 (b) se emite la petición de seguimiento de sincronización.

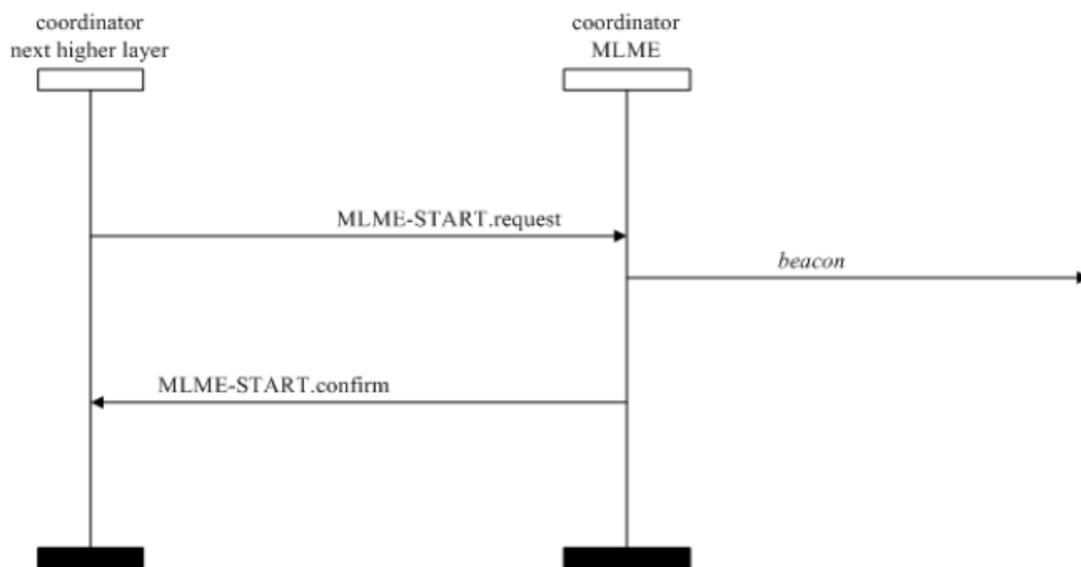


Figura I.7. Secuencia de mensajes para la actualización de la configuración de supertrama.

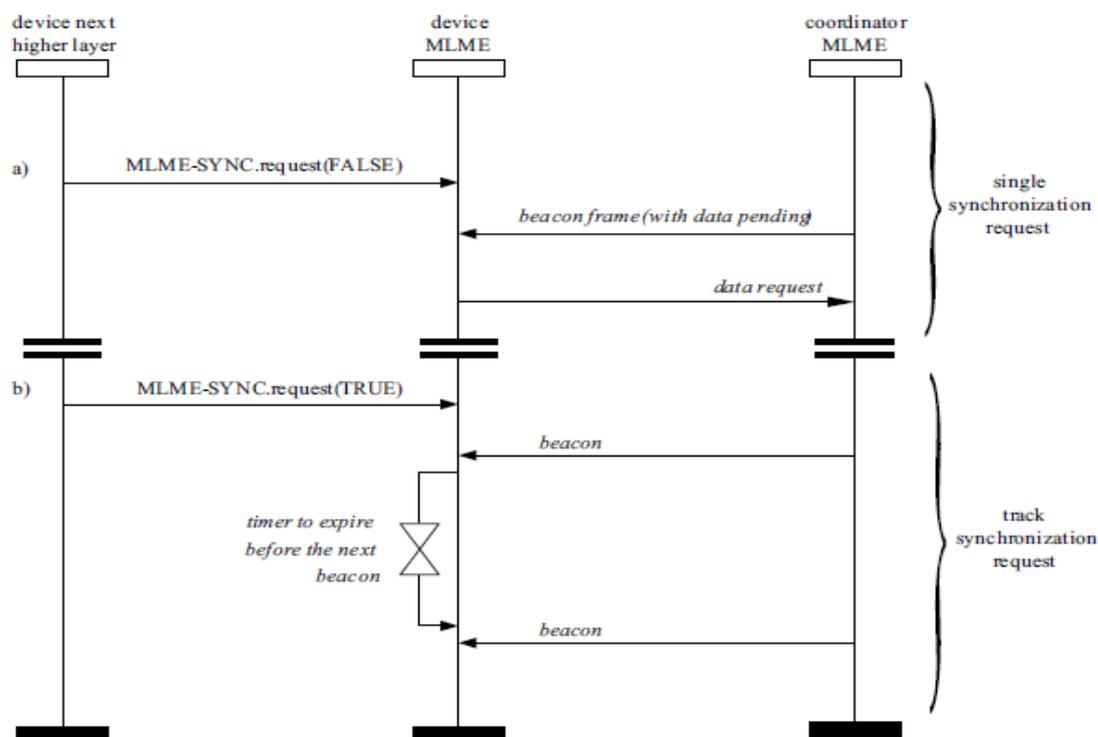


Figura I.8. Secuencia de mensajes para la sincronización con un coordinador en una VPAN con uso de baliza.

### I.2.14. MLME-SAP: Primitivas para la petición de datos de un coordinador

Las primitivas MLME-SAP *polling* definen cómo solicitar datos a un coordinador. Todos los dispositivos proporcionan una interfaz para esas primitivas de sondeo.

#### I.2.14.1. MLME-POLL.request

Estas primitivas proporcionan al dispositivo la petición de datos para el coordinador. Se genera en la capa superior y se emite a su MLME cuando se le van a solicitar datos al coordinador.

#### I.2.14.2. MLME-POLL.confirm

Estas primitivas informan sobre los resultados de una petición de sondeo al coordinador para la obtención de datos. Se genera en la MLME y se emite a su capa superior en respuesta a una primitiva MLME-POLL.request. Si la petición se realiza de forma satisfactoria, el parámetro de estado será igual a SUCCESS, en otro caso se indicará el código de error apropiado.

#### I.2.14.3. Secuencia de mensajes para la petición de datos de un coordinador

La **Figura I.9** muestra la secuencia de mensajes necesaria, incluyendo el comportamiento de la capa del dispositivo y la interfaz, para la petición de datos a un coordinador por parte del dispositivo.

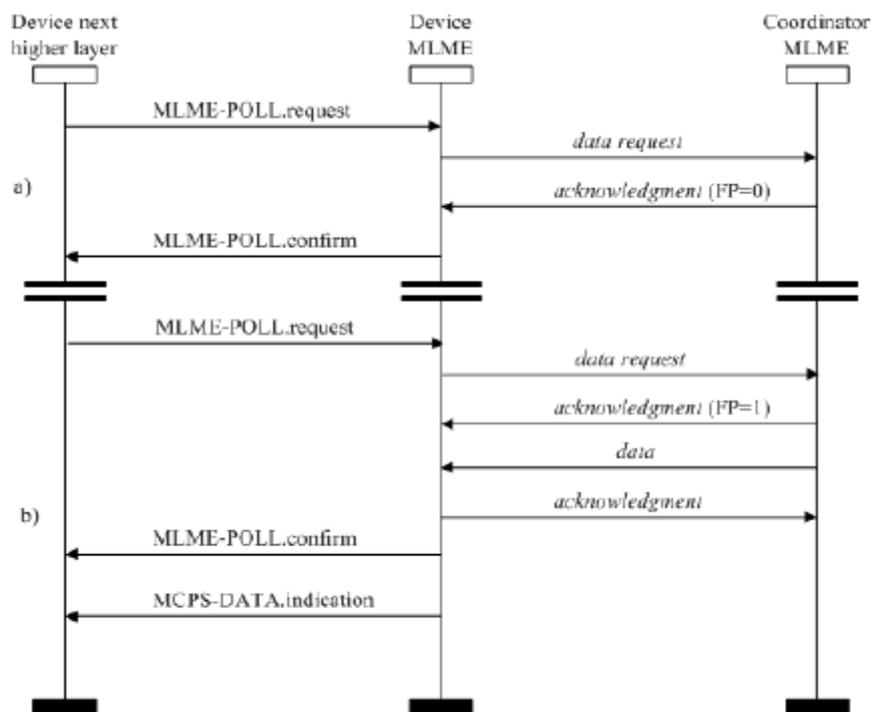


Figura I.9. Secuencia de mensajes para la petición de datos desde un coordinador.

## I.2.15. Primitivas PD-SAP

El PD-SAP soporta el transporte de MPDUs entre una subcapa MAC local y una entidad de capa PHY local. Las primitivas PD-SAP se ilustran en la **Tabla I.4**.

### I.2.15.1. PD-DATA.request

Esta primitiva solicita la transferencia del dato desde la subcapa MAC para formar un PSDU a la entidad PHY local. Admite los siguientes parámetros:

- *pdsuLength*
- *psdu*
- *bandplanID*

## I.2.16. Primitivas PLME-SAP

El PLME-SAP permite el transporte de comandos de gestión entre el MLME o el DME y el PLME. Las primitivas PD-SAP se ilustran en la **Tabla I.5**.



# Anexo II

## FSM para la transmisión de trama de datos con CSMA/CA

---

### 0. Introducción

En este anexo se describe la máquina de estados finitos FSM para la transmisión de trama de datos con CSMA-CA).

En los siguientes apartados describiremos los elementos básicos implicados para el control de la FSM:

- Variables
- Métodos
- Diagrama de estados

### II.1. Variables implicadas en el control de la FSM

Encontramos las siguientes variables:

#### II.1.1. Task

Esta variable es del tipo enumerado “`Ieee802157MacTaskType`” e inicializada en el proceso con el valor “`TP_MCPS_DATA_REQUEST`”, que corresponde al tipo de tarea que maneja la FSM para el proceso de solicitud de transmisión de dato por el MCPS (*medium-access-control common-part sublayer*).

Este tipo enumerado, que está definido en la clase “`Ieee802157Enum.h`”, tiene los siguientes posibles valores:

- `TP_MCPS_DATA_REQUEST = 1`, ←Valor para este proceso
- `TP_MLME_ASSOCIATE_REQUEST`,
- `TP_MLME_ASSOCIATE_RESPONSE`,
- `TP_MLME_DISASSOCIATE_REQUEST`,
- `TP_MLME_ORPHAN_RESPONSE`,
- `TP_MLME_RESET_REQUEST`,
- `TP_MLME_RX_ENABLE_REQUEST`,
- `TP_MLME_SCAN_REQUEST`,
- `TP_MLME_START_REQUEST`,
- `TP_MLME_SYNC_REQUEST`,
- `TP_MLME_POLL_REQUEST`,
- `TP_CCA_CSMACA`,
- `TP_RX_ON_CSMACA`

#### II.1.2. TaskP

Esta variable es del tipo estructura “`taskPending`” y contiene las funciones para la tarea pendiente que esté realizando el MAC. Pueden estar activas distintas tareas a la vez pero no del mismo tipo, esto se controla mediante los campos y funciones de la estructura.

La variable tarea pendiente *taskP* posee los siguientes **campos**:

1. Campo booleano que indica si está activa o no cada tarea
  - a. `bool mcps_data_request;`
  - b. `bool CCA_csmaca;`
  - c. `bool RX_ON_csmaca;`
2. Campo entero para controlar el paso o estado de cada tarea.
  - a. `UINT_8 mcps_data_request_STEP;`
  - b. `UINT_8 CCA_csmaca_STEP;`
  - c. `UINT_8 RX_ON_csmaca_STEP;`
3. Campo de cadena de caracteres que contiene la información de la función que esté realizando en el momento, para la tarea “TP\_MCPS\_DATA\_REQUEST”.
  - a. `char mcps_data_request_frFunc[81];`
4. Campo del tipo enumerado “Ieee802157TxOption” para la tarea “TP\_MCPS\_DATA\_REQUEST”, que posee la información del tipo de transmisión de la trama (directo, indirecto o GTS).
  - a. `Ieee802157TxOption mcps_data_request_TxOption;` Este tipo enumerado, que está definido en la clase “Ieee802157Enum.h”, y tiene los siguientes posibles valores:
 

```
DIRECT_TRANS = 1,
INDIRECT_TRANS = 2,
GTS_TRANS = 3
```
5. Campo que contiene el puntero a la trama para la tarea “TP\_MCPS\_DATA\_REQUEST”
  - a. `Ieee802157Frame* mcps_data_request_pendPkt;`

La variable tarea pendiente *taskP* posee las siguientes **funciones**:

1. Función de inicialización, donde todos las tareas pendientes se desactivan y los pasos o estados se establecen con valor cero.
  - a. `inline void init()`
2. Función que devuelve la referencia al estado de una tarea (*task*), activa o no activa, según el valor booleano correspondiente a cada tipo.
  - a. `bool &taskStatus(Ieee802157MacTaskType task)`
3. Función que devuelve la referencia al paso en el que se encuentra o *STEP*, de una tarea.
  - a. `UINT_8 &taskStep(Ieee802157MacTaskType task)`
4. Función que devuelve un puntero a la cadena de caracteres que indica la función en la que se encuentra una tarea (“Ieee802157MacTaskType”).
  - a. `char *taskFrFunc(Ieee802157MacTaskType task)`

### II.1.3. frmCtrl

Esta variable contiene toda la información de control de la trama que ha recibido de la capa superior para transmitir.

#### **II.1.4. tmpData**

Puntero temporal a la trama que se va a transmitir antes de comenzar el proceso del tipo “Ieee802157DataFrame”, que se crea en el proceso que simula la recepción de la primitiva de solicitud de envío de trama, MCPS\_DATA\_request, y se elimina en el proceso que simula la recepción de la primitiva de indicación de la capa superior MCPS\_DATA\_indication.

#### **II.1.5. txData**

*txData*. Variable global a la cual se le pasa la referencia a la trama de tmpData. Se le da el valor antes de iniciar el proceso (en MCPS\_DATA\_request) y se elimina una vez termina la tarea pendiente del proceso, ya sea exitosa o fallida en los métodos correspondientes taskSuccess o taskFailed.

#### **II.1.6. txCsmaca**

Variable global a la cual se le asigna la referencia a la trama solicitada para transmitir, en este caso será txData, y que a lo largo del proceso CSMA-CA sirve para comprobar qué trama se encuentra actualmente en dicho proceso.

#### **II.1.7. tmpCsmaca**

Variable global a la cual se le asigna la referencia a la trama solicitada para transmitir que será txData como argumento de entrada del método csmacaStart, en el cual se inicia la variable, se utiliza para cálculos temporales al inicio del CSMA-CA y se elimina al terminar el proceso CCA.

#### **II.1.8. waitDataAck**

Variable global de tipo booleano que obtiene el valor true cuando la trama transmitida requiere confirmación ACK y se mantiene dicho valor mientras el proceso se encuentra pendiente de la recepción del ACK.

#### **II.1.9. numDataRetry**

Variable global de tipo entero que contabiliza el número de re-intentos de transmisión de una misma trama de datos, es decir, contabiliza las veces que se realiza este proceso después de cada solicitud.

#### **II.1.10. NB**

Variable global de tipo entero que contabiliza el número de tiempos aleatorios de retardos, backoffs, tras los que el dispositivo intentará acceder y evaluar el canal.

#### **II.1.11. BE**

Variable global del tipo entero que tiene el valor de exponente de backoff, que está relacionada con el número de períodos de retroceso el dispositivo esperará antes de intentar acceder y evaluar el canal.

#### **II.1.12. CW**

Variable global del tipo entero que tiene el valor de la ventana de contención para el proceso de acceso al canal.

### II.1.13. backoffStatus

Variable global de tipo entero que contiene el estado para el proceso de tiempo aleatorio de backoff. Según el valor asignado se indica los siguientes posibles estados:

- 0: no está en tiempo de backoff, o backoff no activo
- 1: backoff exitoso
- 2: backoff fallido
- 99: en tiempo de backoff, o backoff activo

### II.1.14. Otras

- Las siguientes variables de información de supertrama: `bcnRxTime`, `schedBcnRxTime`, `BI`, `phy_mode_min_clk`, `bPeriod`.
- Variables con información temporal.

## II.2. Métodos implicados en el control de la FSM

Algunos de los métodos implicados en el control de la FSM tendrán en cuenta valores del tipo `PHYenum` y `MACenum` para la gestión del proceso. Estos tipos contienen los distintos estados en el que se puede encontrar las diferentes funciones de cada nivel, PHY o MAC. Ambos tipos enumerados están definidos en las clases “*Ieee802157PhyConstEnum.h*” y “*Ieee802157Enum.h*” respectivamente, y sus posibles valores son los siguientes:

#### 1. `PHYenum`:

- `phy_BUSY = 0`,
- `phy_BUSY_RX`,
- `phy_BUSY_TX`,
- `phy_FORCE_TRX_OFF`,
- `phy_IDLE`,
- `phy_INVALID_PARAMETER`,
- `phy_RX_ON`,
- `phy_SUCCESS`,
- `phy_TRX_OFF`,
- `phy_TX_ON`,
- `phy_UNsupport_ATTRIBUTE`

#### 2. `MACenum`:

- `mac_SUCCESS = 0x00`,
- `mac_PAN_at_capacity`,
- `mac_PAN_access_denied`,
- `mac_BEACON_LOSS = 0xe0`,
- `mac_CHANNEL_ACCESS_FAILURE`,
- `mac_DENIED`,
- `mac_DISABLE_TRX_FAILURE`,
- `mac_FAILED_SECURITY_CHECK`,
- `mac_FRAME_TOO_LONG`,

- mac\_INVALID\_GTS,
- mac\_INVALID\_HANDLE,
- mac\_INVALID\_PARAMETER,
- mac\_NO\_ACK,
- mac\_NO\_BEACON,
- mac\_NO\_DATA,
- mac\_NO\_SHORT\_ADDRESS,
- mac\_OUT\_OF\_CAP,
- mac\_PAN\_ID\_CONFLICT,
- mac\_REALIGNMENT,
- mac\_TRANSACTION\_EXPIRED,
- mac\_TRANSACTION\_OVERFLOW,
- mac\_TX\_ACTIVE,
- mac\_UNAVAILABLE\_KEY,
- mac\_UNSUPPORTED\_

Los métodos implicados en el control de la FSM son:

### II.2.1. MCPS\_DATA\_request

Da comienzo a una tarea del tipo “TP\_MCPS\_DATA\_REQUEST” , y llama al método que comprueba si existe tarea pendiente (*taskP*) del mismo tipo e inicia los campos de estado y función de la misma. Una vez comprueba el modo de transmisión aumenta el paso de la tarea el siguiente valor 1, y le da el valor “*csmacaCallback*” al campo función (*mcps\_data\_request\_frFunc*) de dicha tarea.

### II.2.2. checkTaskOverflow

Comprueba que no haya tarea pendiente (*taskP*) del mismo tipo, si existe aparece el error de desbordamiento de tarea, si no hay tarea activa del tipo “TP\_MCPS\_DATA\_REQUEST” inicia los valores de los campos de estado “*mcps\_data\_request\_STEP*” y de función “*mcps\_data\_request\_frFunc*” correspondientes a cero.

### II.2.3. dispatch

Según la función en la que se encuentre la tarea pendiente (*taskP*), y en el caso de ser la transmisión de una trama de datos, cuya opción es la que interesa en este proceso, este proceso llama al método que controla los cambios de estados de la FSM, pasando como argumentos un estado de los tipos *PHYenum* o *MACenum*. Esto lo hace tras asegurarse que la tarea pendiente se encuentra en sincronía con los valores de la llamada al propio método *dispatch*. Esto es, si coinciden su argumento de entrada del tipo cadena de caracteres *frFunc* con la actual función de la tarea pendiente, y si el estado actual de la tarea pendiente del tipo “TP\_MCPS\_DATA\_REQUEST” es activo o valor *true*.

Los valores de las funciones, representan a los métodos implicados en los subprocesos o funciones que el MAC está realizando para la transmisión, y pueden ser los siguientes:

1. *csmacaCallback*. Función que se encarga de volver al proceso CSMA-CA una vez realizada la detección en el canal y actualiza el estado de backoff (*backoffStatus*). La FSM se encuentra en el STEP= 1, y se pasa el estado del canal.

2. `handle_PD_DATA_confirm`. Función que gestiona la recepción de una primitiva de confirmación de datos del nivel PHY y continúa con el proceso si la trama se envió con éxito. La FSM se encuentra en el `STEP=2` y se pasa el estado `phy_SUCCESS`.
3. `handleAck`. Función que gestiona la recepción de un ACK, comprueba que el ACK es el que corresponde a la trama transmitida y si es así, continúa con el proceso. La FSM se encuentra en el `STEP=3` y se pasa el estado `phy_SUCCESS`.
4. `handleAckTimeoutTimer`. Función que gestiona la finalización del tiempo de espera de ACK sin recepción de éste, y continúa con el proceso. La FSM se encuentra en el `STEP=3` y se pasa el estado `phy_BUSY`.

#### II.2.4. FSM\_MCPS\_DATA\_request

Es la máquina de estados finita que maneja los estados para la solicitud de transmisión de trama de datos según sea transmisión directa, indirecta o por GTS. Los estados son los valores controlados por la variable miembro de las tareas llamada “`taskStep`” de la tarea activa.

En el caso concreto de transmisión directa de una trama de datos, el tipo de tarea pendiente que se activa es la llamada `TP_MCPS_DATA_request` (tarea pendiente de solicitud MCPS de transmisión de trama de datos).

Dependiendo del valor del estado o paso de la tarea pendiente, `STEP`, y según el argumento de entrada de estado del proceso, realiza ciertas acciones en cada caso y actualiza el valor de estado o paso de la FSM.

El argumento de entrada de estado del proceso, puede ser de los tipos `PHYenum` y `MACenum`, cuyo valor determinará el cambio de `STEP` o estado en la FSM.

Los posibles estados y la relación de entrada y salida entre ellos son los siguientes:

- 1) **Step=0**: Estado en el que se encuentra la tarea al inicializarse. No ocurre nada en este estado, y no tiene condiciones de entradas y salidas, es decir, se mantiene en este estado hasta que otro método modifique el campo de estado de la tarea pendiente (`mcps_data_request_STEP`) a través de la variable “`taskStep`”.
- 2) **Step=1**: la FSM asigna el valor 1 por primera vez tras la solicitud `MCPS_DATA_request` (solicitud MCPS de transmisión de trama de datos) y **la FSM se encuentra en este estado cuando finaliza el proceso CCA**. En este estado se evalúa el estado del canal tras realizar el CCA, si éste está libre se solicita encender el transmisor y si está ocupado se vuelve al comienzo del proceso CSMA/CA (`csmacaResume`).

**Entradas:** Desde el método `MCPS_DATA_request`. Desde la condición de canal ocupado del propio estado.

**Salida:** Si el canal está libre para transmitir, pasa al siguiente estado **Step=2**

- 3) **Step=2**: Se le asigna este valor en el estado anterior y **la FSM se encuentra en este estado cuando recibe la primitiva de confirmación de transmisión de trama de datos de la capa PHY**. En él se evalúa si se requiere o no acuse de recibo, en el caso de que no se requiera ACK da por finalizado el proceso con éxito y finaliza la tarea pendiente, si la trama requiere ACK solicita encender el receptor.

**Entrada:** Desde la condición del estado anterior de canal libre.

**Salidas:** Si requiere ACK pasa al estado **Step=3**. Si no requiere ACK, sale de la FSM.

- 4) **Step=3:** Se asigna este valor en el estado anterior y sólo cuando requiere ACK la trama enviada, y **la FSM se encuentra en este estado cuando el MAC ha recibido el ACK o cuando ha finalizado el tiempo de espera de ACK**. En él se evalúa si se ha recibido el ACK dentro del tiempo de espera de ACK o hay que reintentar la transmisión. Si se recibe correctamente dentro del tiempo de espera da por finalizado el proceso con éxito y finaliza la tarea pendiente. Si no se recibe con éxito se vuelve al comienzo del proceso CSMA/CA (*csmacaResume*) en el caso de que no haya superado el número máximo de reintentos, si ha sido superado, entonces da por finalizado el proceso con fallo y finaliza la tarea pendiente.

**Entrada:** Desde la condición del estado anterior de ACK requerido.

**Salidas:** Si recibe ACK sale de la FSM. Si no recibe ACK y supera el número de reintentos, sale de la FSM. Si no recibe ACK y el número de reintentos es menor al máximo pasa al estado **Step=1**.

### II.2.5. taskFailed

Es una de las dos posibles salidas del proceso y de la FSM. En el caso de trama de datos, se encarga de eliminar la referencia en *txData*, simula el envío de primitiva de confirmación *MCPS\_DATA\_confirm*, y solicitar otro paquete del módulo cola, a no ser que se le indique mediante el argumento de tipo booleano *csmacaRes* con el valor *true*, que reinicie el proceso CSMA/CA.

### II.2.6. taskSuccess

Es una de las dos posibles salidas del proceso y de la FSM. En el caso de trama de datos, se encarga de eliminar la referencia en *txData*, simula el envío de primitiva de confirmación *MCPS\_DATA\_confirm*, y solicitar otro paquete del módulo cola.

## II.3. Diagrama de estados de la FSM

En la **Figura II.1** se detalla el diagrama de estados de la FSM, indicando las variables y métodos involucrados, así como las transiciones entre los posibles estados dados por el valor de la variable STEP.

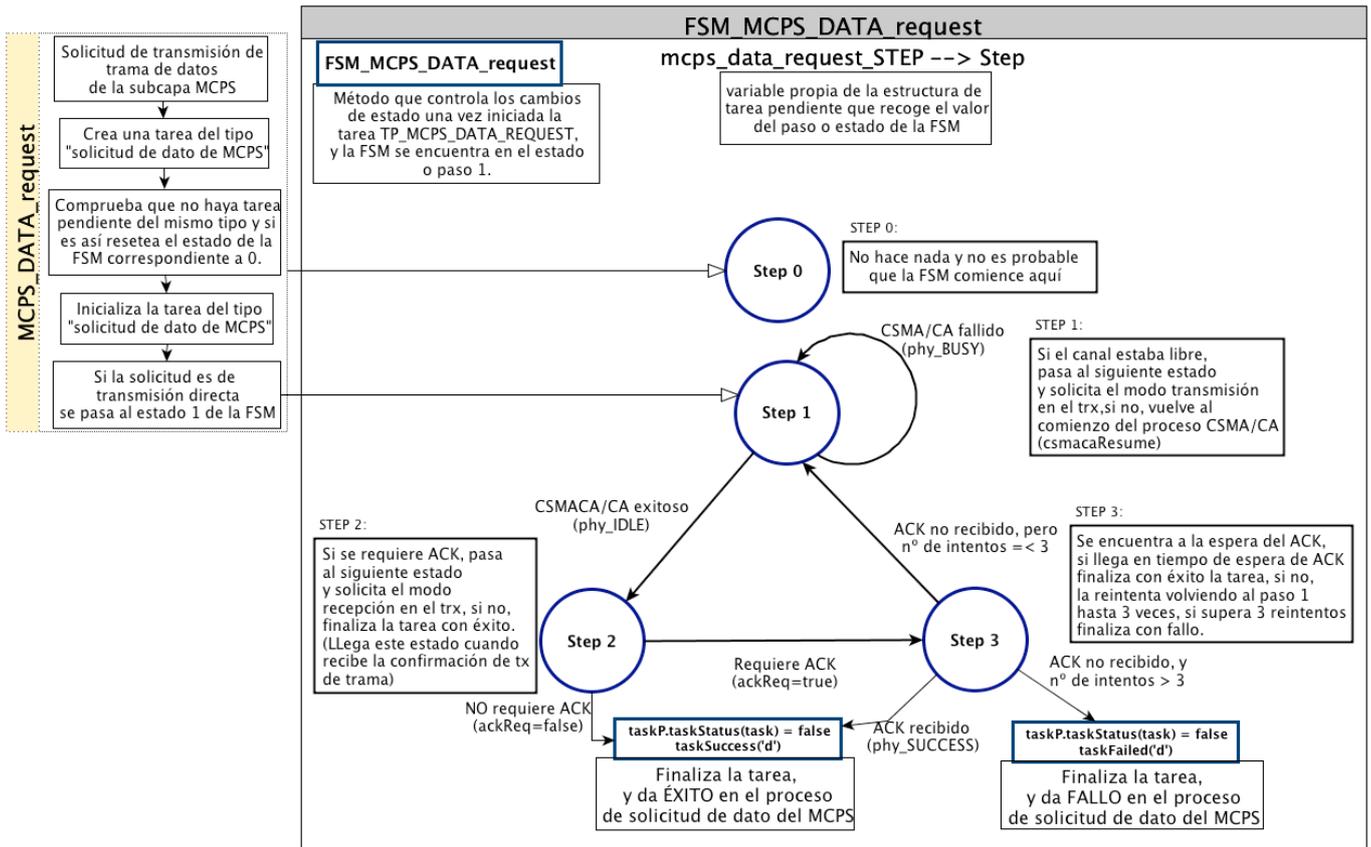


Figura II.1. Diagrama de estados de la FSM.

# Anexo III

## Algoritmo de acceso al canal de la norma IEEE.802.15.7

---

### 0. Introducción

El comportamiento del código implementado para el algoritmo de acceso al canal en INETMANET se corresponde con el algoritmo descrito en la norma IEEE.802.15.4. En este anexo se analizan las diferencias en el algoritmo de acceso al canal con respecto al modelo necesario para VLC, detallando las deficiencias a cubrir.

### III.1. Algoritmo de acceso aleatorio

El algoritmo de acceso aleatorio ranurado (**Figura III.1**) deberá ser usado ante de la transmisión de tramas de datos o comandos MAC transmitidas dentro del CAP, a menos que la trama pueda ser transmitida rápidamente después de la confirmación de un comando de petición de dato. Ninguno de los algoritmos de acceso aleatorio deberá usarse para la transmisión de tramas de balizamiento en una VPAN con baliza habilitada, tramas de confirmación o tramas de datos transmitidas en el CFP.

Si las balizas periódicas están siendo usadas en la VPAN, la MAC deberá emplear la versión ranurada del algoritmo de acceso aleatorio para las transmisiones en el CAP de la supertrama. Al contrario, si no se usan balizas periódicas en la VPAN o si una baliza no pudiera ser localizada en una VPAN con baliza, la MAC deberá transmitir usando la versión no ranurada del algoritmo de acceso aleatorio.

En ambos casos, el algoritmos se implementa usando unidades de tiempo llamadas periodos de backoff, donde un periodo backoff deberá ser igual a  $aUnitBackoffPeriod$  relojes ópticos.

En el acceso aleatorio ranurado, los límites del periodo de backoff de cada dispositivo en la VPAN deberán estar alineados con los límites de slot de la supertrama del coordinador, es decir, el comienzo del primer periodo de backoff de cada dispositivo se alinea con el comienzo de la transmisión de la baliza.

En el acceso aleatorio ranurado, la MAC deberá asegurarse de que la PHY comienza todas sus transmisiones en el límite de un periodo de backoff. En el acceso aleatorio no ranurado, los periodos de backoff de un dispositivo no se relacionan en tiempo a los periodos de backoff de otros dispositivos en la VPAN.

Cada dispositivo deberá mantener dos variables para cada intento de transmisión:  $NB$  y  $BE$ .  $NB$  es el número de tiempos del algoritmo de acceso requerido para el backoff al intentar la transmisión actual; este valor deberá ser inicializado a cero antes de cada nuevo intento de transmisión.

La variable  $BE$  es el exponente de backoff, que tiene que ver con cuantos periodos de backoff deberá esperar un dispositivo antes de intentar acceder/evaluar un canal.  $BE$  deberá ser inicializado al valor de  $macMinBe$ .

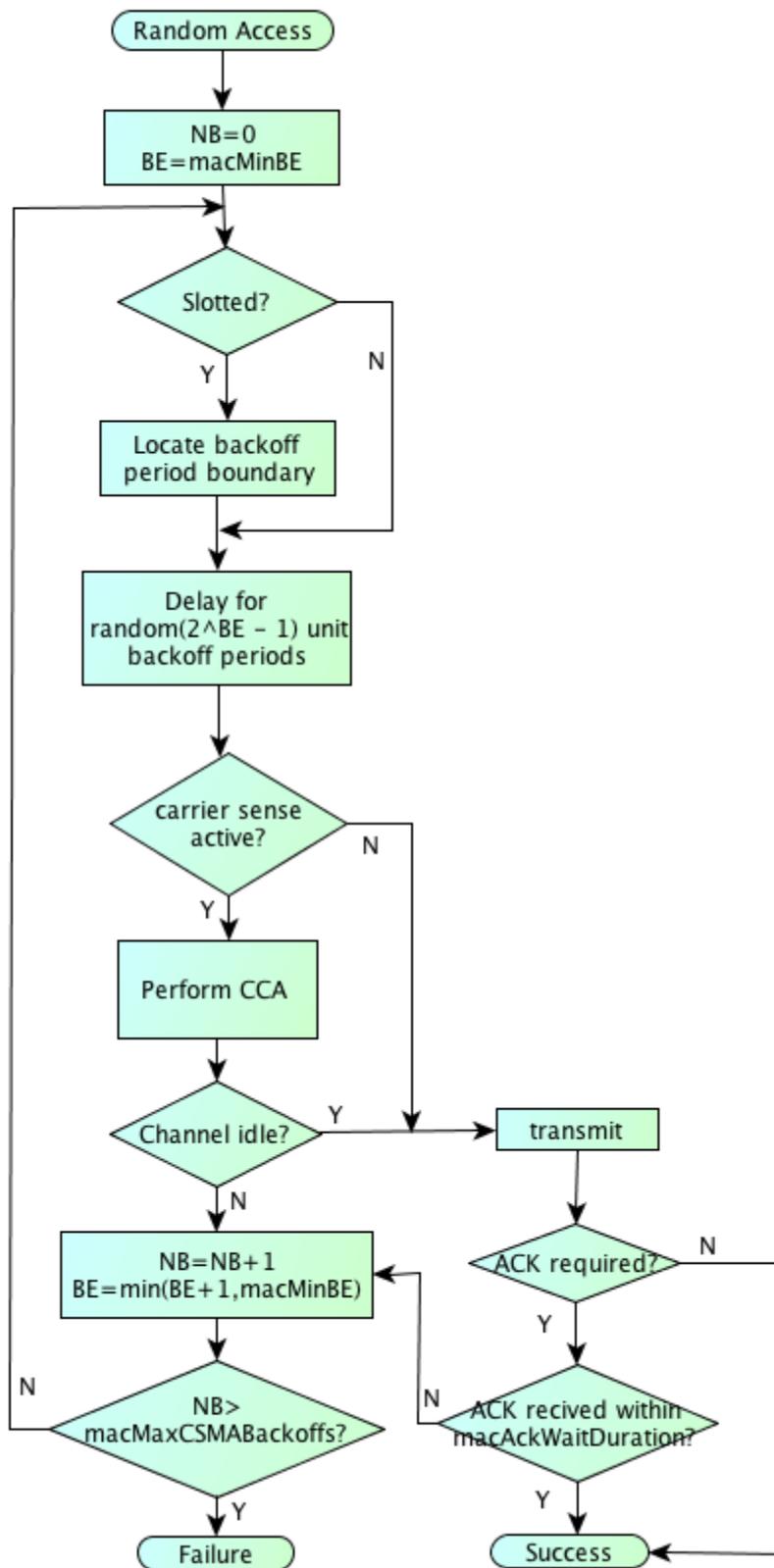


Figura III.1. Diagrama de flujo del acceso aleatorio.

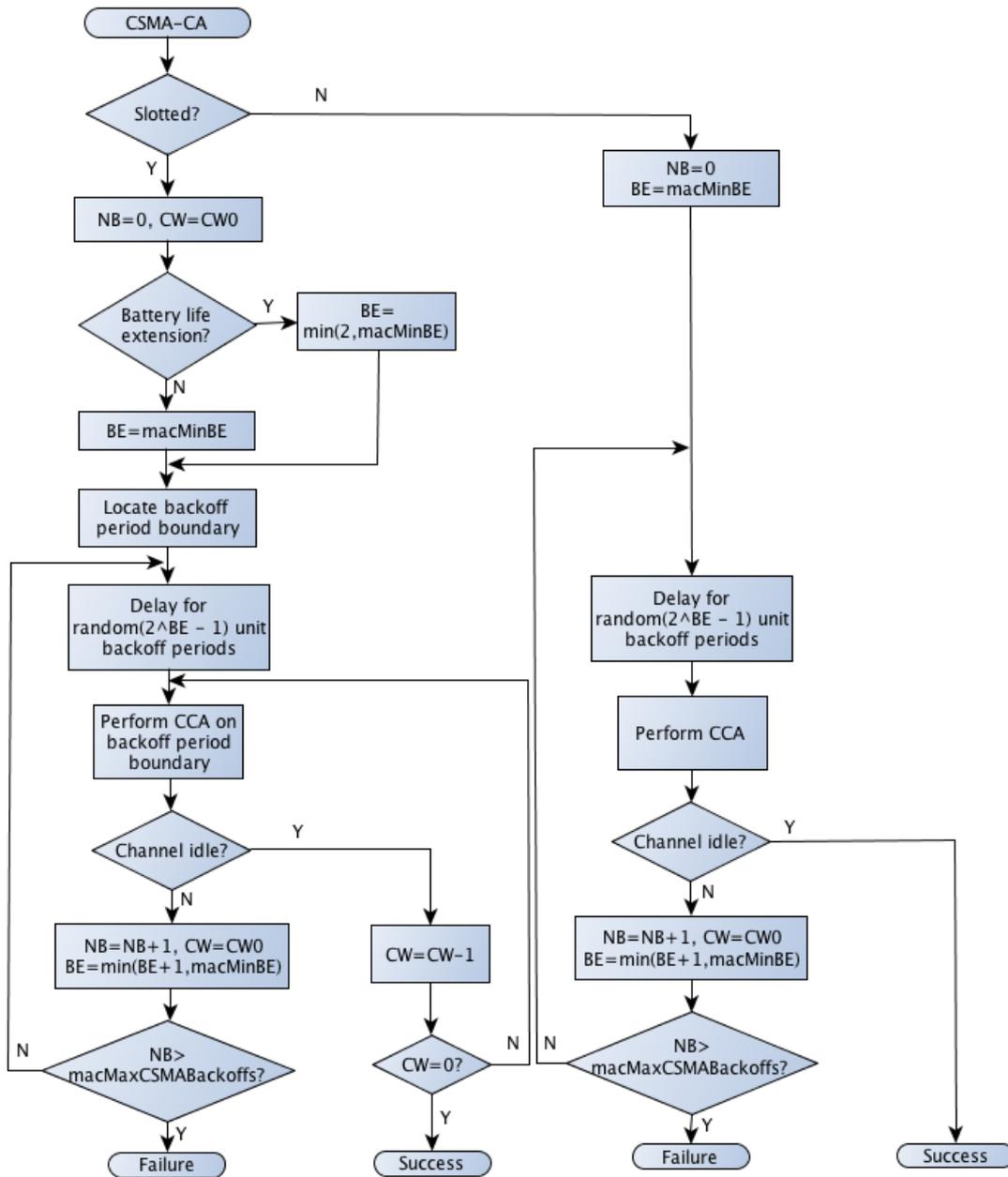


Figura III.2. Diagrama de flujo del CSMA-CA.

La subcapa MAC deberá primero inicializar  $NB$  y  $BE$  para el acceso aleatorio ranurado luego localizar el límite del siguiente periodo de backoff. La MAC deberá retrasar un número aleatorio para completar los periodos de backoff en el rango 0 to  $2BE$  hasta el comienzo del CAP en la siguiente supertrama y aplicar un retardo de backoff aleatorio adicional antes de evaluar si puede continuar de nuevo.

Si la CCA está activa y el canal está valorado como ocupado, la subcapa MAC deberá incrementar tanto el  $NB$  como el  $BE$  en uno, asegurando que el  $BE$  deberá ser no más que  $macMaxBE$ . Si el valor de  $NB$  es menor o igual que el  $macMaxRABackoffs$ , el algoritmo de acceso deberá volver a realizar un backoff aleatorio. Si el valor de  $NB$  es mayor que  $macMaxRABackoffs$ , el algoritmo de acceso deberá terminar con el estado de fallo de acceso al canal.

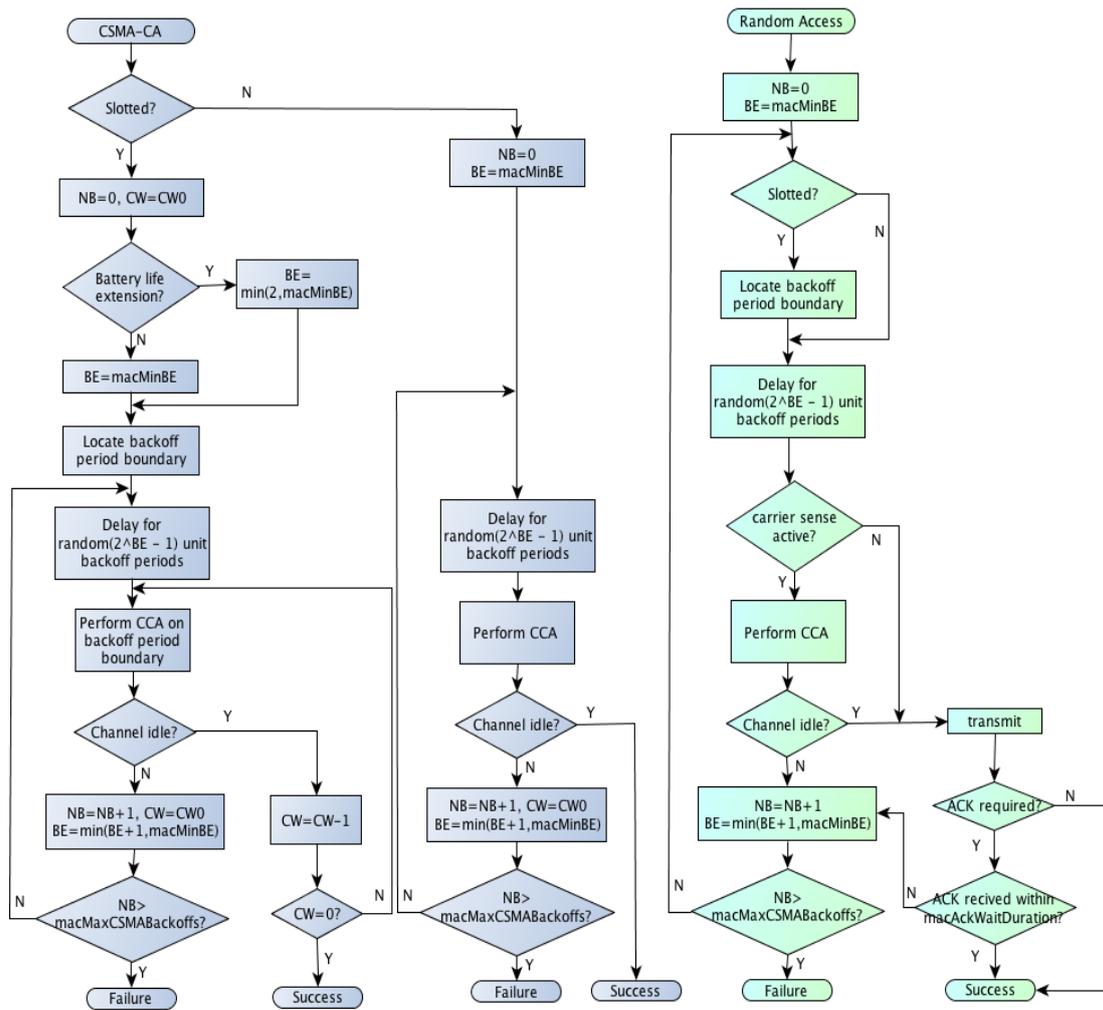


Figura III.3. Comparación de algoritmos.

Aparecen dos errores en el diagrama del estándar, uno es la configuración del parámetro BE para cada reintento, debería ser  $BE = \min(BE+1, macMaxBE)$  y el otro sería la consideración de decisión de reintento, la cual debería ser  $NB > macMaxMABackoffs$ .

### III.2. Diferencias con el algoritmo de acceso al canal de la norma IEEE.802.15.4

Los diagramas de flujo de cada estándar (Figura III.1 y Figura III.2) nos permiten ver de forma clara las diferencias entre los distintos algoritmos (Figura III.3).

El algoritmo descrito en el estándar LR-WPAN, visto en la Figura III.2, divide el proceso en dos ramas desde el principio, según si el acceso al canal es ranurado o no ranurado. Realizando la comparación entre ambas normas, podemos observar que el algoritmo del estándar VLC, se aproxima mucho a la rama de acceso no ranurado del proceso seguido por la norma IEEE.802.15.4, habiendo que añadir la opción de localización de límite de periodo de backoff en el caso en que sea ranurado. Además el diagrama en VLC incluiría a dicha rama: la opción de saltarse el proceso CCA; y el flujo a partir de la transmisión, es decir, no considera éxito hasta que haya recibido la confirmación de la trama cuando ésta es requerida.

En resumen, las principales diferencias serían las enumeradas a continuación:

1. En VLC no hay ventana de contención en el algoritmo.
2. En VLC existe la posibilidad de deshabilitar la escucha de canal (CCA) y transmitir “a ciegas”.
3. En VLC los parámetros de inicialización son comunes para todos los casos, y se inicializan antes de comprobar el tipo de acceso.
4. En VLC el algoritmo incluye el proceso de comprobación (ACK) para considerar con éxito o no el acceso al canal, si éste es requerido.
5. En VLC el algoritmo aumenta el número de intentos de backoff también cuando no recibe ACK correctamente, si éste es requerido.

La principal diferencia, y que no se observa claramente en los gráficos, es que para el estándar LR-WPAN, no recibir un ACK cuando la trama transmitida lo requería es un reintento del proceso completo, es decir desde cero incluyendo la inicialización de los valores NB y BE, sin embargo, en VLC, el reintento es dentro del mismo proceso de acceso al canal sin volver a inicializar dichos valores.

### III.3. Implementación en WPAN de INETMANET del CSMA-CA del IEEE.802.15.4

Antes de entrar a describir los métodos y funciones implicados en el proceso CSMA-CA, se describe a continuación la estructura C++ “taskPending” para las tareas pendientes implementada en la clase “Ieee802157Mac.h”:

- mcps\_data\_request = false;
- mcps\_data\_request\_STEP = 0;
- CCA\_csmaca = false;
- CCA\_csmaca\_STEP = 0;
- RX\_ON\_csmaca = false;
- RX\_ON\_csmaca\_STEP = 0;

La estructura “taskpending” tiene un método de llamada inicial que asigna los siguientes valores a las variables de tareas implementadas en la FSM de la clase

Además contiene la función “&taskStatus(Ieee802157MacTaskType task)” que devuelve un valor booleano de estado de tarea según el tipo de tarea con el que se llame a la función. Las tareas pueden ser las siguientes:

1. TP\_MCPS\_DATA\_REQUEST por el cual devolverá el valor booleano que contenga la variable global “mcps\_data\_request”
2. TP\_CCA\_CSMACA por el cual devolverá el valor booleano que contenga la variable global “CCA\_csmaca”
3. TP\_TP\_RX\_ON\_CSMACA por el cual devolverá el valor booleano que contenga la variable global “RX\_ON\_csmaca”
  - Por defecto “RX\_ON\_csmaca”

También contiene la función “&taskStep(Ieee802157MacTaskType task)” que devuelve un valor entero del paso en el que se encuentra cada tarea según el tipo de tarea con el que se llame a la función. Las tareas pueden ser las siguientes:

1. TP\_MCPS\_DATA\_REQUEST por el cual devolverá el valor entero que contenga la variable global “mcps\_data\_request\_STEP”

2. TP\_CCA\_CSMACA por el cual devolverá el valor entero que contenga la variable global “CCA\_csmaca\_STEP”
3. TP\_TP\_RX\_ON\_CSMACA por el cual devolverá el valor entero que contenga la variable global “RX\_ON\_csmaca\_STEP”
  - Por defecto “RX\_ON\_csmaca\_STEP”

Y por último contiene la función “&taskFrFunc(Ieee802157MacTaskType task)” que devuelve una cadena de caracteres con el nombre de la función de cada tarea según el tipo de tarea con el que se llame a la función. Las tareas pueden ser las siguientes:

1. TP\_MCPS\_DATA\_REQUEST por el cual devolverá el valor entero que contenga la variable global “mcps\_data\_request\_frFunc”
  - Por defecto devuelve “0”.

### III.3.1. Métodos y funciones implicadas

Los métodos y funciones que están implicados en el proceso completo son los siguientes:

1. **csmacaEntry**: Según el tipo de paquete que el nodo va a transmitir inicializa los parámetros de espera de acuse de recibo y número de intentos correspondiente y llama al método “csamacaResume”.
2. **csmacaResume**: Primero se asegura que no esté en periodo de backoff o durante una transmisión, y según el buffer que contenga la trama (txBcnCmd, txBcnCmdUpper o txData) y comprobando a su vez que no tiene ACK pendiente de ninguno de los tres tipos (BcnCmd, BcnCmdUpper o Data), inicializa el estado de backoff, recoge en la variable global “ackReq” si el paquete requiere confirmación o no, guarda la trama en el buffer de transmisión por CSMA-CA (“txCsmaca”) y llama al método “csmacaStart” indicando que es el primer intento, e incluyendo en la llamada la trama y la información recogida en “ackReq”. Además, en el caso de ser una trama de datos, actualiza el tipo de tarea a la función “csmacaCallBack” y al paso 1 de la FSM.
3. **csmacaStart**: Si se llama cuando existe una trama de acuse de recibo pendiente de transmitir anula el estado de backoff (valor a 0) y la trama que está en el buffer temporal para el proceso CSMA-CA (“tmpCsmaca”). Si no, primero se asegura que no haya un periodo de backoff planificado.

Si es el primer intento de proceso CSMA-CA, llama a la función “csmacaReset”, se asegura de que esté vacío el buffer temporal para el proceso CSMA-CA, en cuyo caso lo actualiza con la trama recibida y recoge en la variable “csmacaAckReq” si requiere o no trama de confirmación.

Si el nodo está configurado con baliza habilitada calcula el instante en que la última baliza debió haberse recibido. A posteriori elige un valor aleatorio para el tiempo de backoff, ajustándolo para que éste dentro del periodo de contención (CAP) y no antes, mediante la llamada a la función “csmacaAdjustTime”.

Además, en el caso de que tenga baliza habilitada, ajusta el tiempo de backoff según el límite del final del beacon o baliza mediante la llamada a la función “csmacaLocateBoundary”.

Por último comprueba que el proceso de transmisión de la trama mediante CSMA-CA se puede realizar durante el periodo de contención de la supertrama actual mediante la función “csmacaCanProceed”, y si es así llama al método “startBackoffTimer” para planificar dicho tiempo de backoff aleatorio.

4. **startBackoffTimer:** Si ya existe un backoff planificado cancela dicho evento, si no planifica el automensaje para un nuevo evento de tiempo de backoff.
5. **handleBackoffTimer:** En el instante de backoff planificado se inicia el método, el cual envía la primitiva correspondiente a la capa PHY para iniciar el receptor y a su vez pasa el estado de las tareas que maneja la FSM del nodo a “TP\_RX\_ON\_CSMACA”.
6. **handle\_PLME\_SET\_TRX\_STATE\_confirm:** Maneja la primitiva PHY de confirmación de estado del tranceptor. En el caso en que la capa PHY haya devuelto el estado de la solicitud con éxito y además el estado de las tareas sea “TP\_RX\_ON\_CSMACA”, éste estado de tarea es establecido a FALSE, es decir, se considera finalizada la tarea, y se llama al método “csmaca\_handle\_RX\_ON\_confirm” con la función solicitada a la capa PHY (en este caso phy\_RX\_ON).

En el caso de que la capa además de devolver el estado de la solicitud con éxito la solicitud ha sido phy\_TX\_ON, y existe una trama de confirmación de dato en el buffer de transmisión “txAck” se ajusta el retardo para la transmisión del tiempo límite de ACK mediante la función “csmacaLocateBoundary” y se llama al método “startTxAckBoundTimer” o directamente al “handleTxAckBoundTimer” si el retardo es cero.

7. **csmaca\_handle\_RX\_ON\_confirm:** Si el estado de la capa física fue ocupado transmitiendo volverá se vuelve a establecer la tarea “TP\_RX\_ON\_CSMACA” al valor TRUE, a espera de que la capa física confirme que ha encendido el receptor. Si la solicitud no es exitosa vuelve a solicitar encender el receptor llamando al método “handleBackoffTimer”.

Cuando el estado de la capa física es el correspondiente a haber encendido el receptor con éxito, establece a FALSE el paso de tarea “TP\_RX\_ON\_CSMACA”, además si el nodo tiene baliza habilitada, se vuelve a localizar el límite de backoff según los periodos de backoff de la supertrama con el método “csmacaLocateBoundary”, y calculando el retardo con el método “startDeferCCATimer”, una vez está ajustado se solicita la escucha del canal a la capa PHY mediante una primitiva, llamando al método “PLME\_CCA\_request” y se actualiza el estado de tarea al paso “TP\_CCA\_CSMACA”.

8. **PLME\_CCA\_request:** Construye la primitiva PLME\_CCA\_request y la envía a la capa física.
9. **handle\_PLME\_CCA\_confirm:** Cuando llega la primitiva PHY de confirmación de CCA, si efectivamente las tareas de la FSM estaban en dicho paso “TP\_CCA\_CSMACA”, lo establece a FALSE, y según el estado del canal libre u ocupado. Para el caso en que el canal está libre, repite el

proceso de CCA hasta que la ventana de contención, recogida en la variable global CW, sea cero y luego llama al método “csmacaCallback” con el estado IDLE. Para el caso en el que el canal está ocupado, reintenta la escucha con un nuevo tiempo aleatorio de backoff llamando al método “csmacaStart” siempre que no se haya superado el valor máximo establecido por la constante macMaxCSMABackoffs, controlado por la variable global NB la cual contiene el número de intentos de backoffs.

10. **csmacaCallback:** Se asegura que el buffer de trama de transmisión no está vacío y que no está a la espera de confirmación de trama, y en ese caso si el estado del canal era IDLE pasa el estado de backoff a exitoso y llama al método para ejecución de tareas “dispatch” con dicho estado y la función actual (establecida al inicio del proceso CSMA-CA en el método “csmacaResume” al valor “csmacaCallback”).
11. **dispatch:** Llama a la FSM pasándole el valor correspondiente a la función de tarea actual.
12. **FSM\_MCPS\_DATA\_request:** Máquina de estados finita que maneja los estados para la solicitud de transmisión de trama de datos según sea transmisión directa, indirecta o por GTS. En el caso de transmisión directa existen 4 pasos, desde 0 a 3.

Una vez haya comenzado el proceso CSMA-CA estará en el paso 1, en el cual si el canal está libre pasará la función de tareas a “handle\_PD\_DATA\_confirm”, solicitará a la capa PHY que encienda el transmisor mediante el método “PLME\_SET\_TRX\_STATE\_request” y se aumenta el paso al valor 2. Si el canal está ocupado, llamará al método “csmacaResume”.

En el paso 2, si la trama transmitida, requiere acuse de recibo, se aumenta el paso al valor 3, la función de tarea pasa a ser “handleAck”, se solicita a la capa física encender el receptor, se llama al método “startAckTimeoutTimer” y se establece al valor TRUE la variable “wait-DataAck”.

En el caso de no requerir acuse de recibo, se resetea el transceptor mediante la llamada al método “resetTRX”, se llama al método de tarea con éxito “taskSuccess”, terminando aquí el proceso de transmisión de paquete de datos mediante CSMA-CA con éxito, con lo que se establece el estado de tarea “TP\_MCPS\_DATA\_REQUEST” al valor FALSE.

En el paso 3, si se ha recibido con éxito el ACK, es decir, dentro del tiempo de timeout programado, se resetea el transceptor mediante la llamada al método “resetTRX”, se llama al método de tarea con éxito “taskSuccess”, terminando aquí el proceso de transmisión de paquete de datos mediante CSMA-CA con éxito, con lo que se establece el estado de tarea “TP\_MCPS\_DATA\_REQUEST” al valor FALSE.

Si no se recibe el ACK durante dicho tiempo, si no se ha superado el máximo número de reintentos limitado por el valor de la constante aMaxFrameRetries, entonces se vuelve al paso 1, la función de tareas pasa a ser “csmacaCallback”, la variable “waitDataAck” se establece al valor FALSE y se llama a la función “csmacaResume” para retransmisión del paquete de datos no confirmado.

Si se ha superado el valor máximo de reintentos, entonces se resetea el transceptor mediante la llamada al método “resetTRX”, se llama al método de tarea fallida “taskFailed”, terminando aquí el proceso de transmisión de paquete de datos mediante CSMA-CA sin éxito, con lo que se establece el estado de tarea “TP\_MCPS\_DATA\_REQUEST” y la variable “waitDataAck” se establecen al valor FALSE.

13. **startAckTimeoutTimer:** Si ya existe un timeout planificado cancela dicho evento, si no planifica el automensaje para un nuevo evento de timeout para recibo de ACK.
14. **handleAckTimeoutTimer:** Si no hay trama de transmisión pendiente, llama a la función de error con el valor “handleAckTimeoutTimer” y al método “dispatch”.



# Anexo IV

## Gestión de transacción. Modo indirecto

---

### 0. Introducción

En este anexo se describe el procedimiento descrito en el estándar IEEE802.15.7/VLC para la gestión de transacciones entre dispositivos.

### IV.1. Gestión de transacción

Las transacciones pueden ser inicializadas desde los propios dispositivos así como desde el coordinador. Es decir, tanto el coordinador necesita indicar en su baliza cuándo los mensajes están pendientes para los dispositivos como los propios dispositivos necesitan sondear al coordinador para determinar si ellos tienen algún mensaje pendiente. Tales transferencias son llamadas transmisiones indirectas.

El coordinador empezará manejando una transacción en la recepción de una petición de transmisión indirecta tanto mediante la primitiva `MCPS-DATA.request` o mediante una petición del `MLME` para enviar un comando MAC instigado por una primitiva desde la siguiente capa superior, tal como la primitiva `MLME-ASSOCIATE.response`. Al finalizar tal transacción, la subcapa MAC indicará un valor de estado a la siguiente capa superior. Si una primitiva de petición instigó la transmisión indirecta, la correspondiente primitiva de confirmación será usada para comunicar el valor de estado apropiado. De manera inversa, si una primitiva de respuesta instigó la transmisión indirecta, la primitiva `MLME-COMM-STATUS.indication` puede ser relacionada con su primitiva de respuesta correspondiente examinando el campo de dirección de destino (*Destination Address*).

La información contenida en la petición de transmisión indirecta forma una transacción, y el coordinador será capaz de almacenar al menos una transacción. En la recepción de una petición de transmisión indirecta, si no hay capacidad para almacenar otra transacción, la MAC indicará a la siguiente capa superior un estado de `TRANSACTION_OVERFLOW` en la primitiva correspondiente apropiada.

Si el coordinador es capaz de almacenar más de una transacción, se asegurará de que todas las transacciones para el mismo dispositivo sean enviadas en el orden en el cual han llegado a la subcapa MAC. Para cada transacción enviada, si existen otras del mismo dispositivo, la MAC establecerá el subcampo de trama pendiente a valor 1, indicando el dato pendiente adicional.

Cada transacción persistirá en el coordinador durante al menos un tiempo indicado por el parámetro `macTransactionPersistenceTime`. Si la transacción no se ejecuta satisfactoriamente por el dispositivo apropiado dentro de este tiempo, la información de transacción será descartada y la MAC indicará a la siguiente capa superior un estado de `TRANSACTION_EXPIRED` en la primitiva correspondiente apropiada. Con el fin de ser ejecutada de forma satisfactoria o con éxito, se recibirá confirmación si ésta fue requerida.

Si la transacción fue exitosa, la información de transacción será descartada, y la MAC indicará a la siguiente capa superior un estado de **SUCCESS** en la apropiada primitiva correspondiente.

Si el coordinador transmite balizas, hará una lista de las direcciones de los dispositivos en los que cada transacción está asociada en el campo lista de direcciones (*Address List*) e indicará el número de direcciones en el campo de especificación de dirección pendiente (*Pending Address Specification*) de la trama de balizamiento.

Si el coordinador es capaz de almacenar más de siete transacciones pendientes, las indicará en su baliza en la forma “*primero que llega - primero servido*”, asegurándose que la trama de baliza contiene al menos siete direcciones. Para las transacciones que requieren GTS, el coordinador no añadirá la dirección del destinatario de la lista de direcciones pendientes en la trama de baliza. En su lugar, transmitirá la dirección en el GTS asignado para el dispositivo.

En una VPAN con baliza habilitada, si hay una dirección pendiente para la dirección de multidifusión, el subcampo de trama pendiente del campo de control de trama en la trama de baliza se establecerá a uno, y el mensaje pendiente se transmitirá inmediatamente a continuación de la baliza usando el algoritmo de acceso aleatorio no ranurado. Si hay un segundo mensaje pendiente para la dirección de multidifusión, su transmisión se retardará hasta la siguiente supertrama. Se permitirá enviar indirectamente solo un mensaje de multidifusión por supertrama.

En una VPAN con baliza habilitada, un dispositivo que recibe una baliza que contiene su dirección en la lista de transacciones pendientes tratará de extraer el dato del coordinador. En una VPAN sin baliza habilitada, un dispositivo intentará extraer el dato del coordinador al recibir la primitiva **MLME-POLL.request**.

El procedimiento para la extracción del dato pendiente se describe más adelante. Si un dispositivo recibe una baliza con el subcampo de trama pendiente establecido a uno, dejará su receptor habilitado *macMaxFrame-TotalWaitTime* relojes ópticos para recibir el dato de multidifusión del coordinador.

## IV.2. Extracción de un dato pendiente del coordinador

Un dispositivo en una VPAN con baliza habilitada puede determinar si hay alguna trama pendiente para el dispositivo examinando el contenido de la trama de baliza recibida. Si la dirección del dispositivo está en el campo de lista de direcciones de la trama de baliza y el parámetro *macAutoRequest* tiene el valor **TRUE**, el MLME del dispositivo podría enviar un comando de solicitud de datos al coordinador durante el CAP con el subcampo solicitud de confirmación del campo de control de trama activado; la única excepción a esto es si la trama de baliza se recibe mientras se realiza una exploración del medio de forma activa o pasiva.

Hay dos otros casos por los cuales el MLME podría enviar un comando de solicitud de datos al coordinador. El primer caso es cuando el MLME recibe la primitiva de solicitud **MLME-POLL.request**. En el segundo caso, un dispositivo puede enviar un comando de solicitud de datos *macResponseWaitTime* relojes ópticos después de una confirmación a una trama de comando solicitada, tal como durante el proceso de asociación. Si la solicitud de dato está destinada al coordinador, la información de dirección de destino podría ser omitida.

Al recibir el coordinador un comando de solicitud de dato de forma exitosa, éste deberá enviar una trama de confirmación, corroborando así su recepción. Si el coordinador

tiene suficiente tiempo para determinar si el dispositivo tiene efectivamente una trama pendiente antes de enviar dicha trama de confirmación, deberá establecer el subcampo de trama pendiente del campo de control de trama de ésta en consecuencia para indicar si hay una trama pendiente para el dispositivo actualmente.

Al recibir una trama de confirmación con el subcampo de trama pendiente a cero, el dispositivo deberá concluir que no hay dato pendiente en el coordinador para él.

Al recibir una trama de confirmación con el subcampo de trama pendiente al valor uno, un dispositivo deberá habilitar su receptor durante al menos *macMaxFrameTotalWaitTime* relojes ópticos del CAP en una VPAN con baliza habilitada, para recibir la correspondiente trama de datos del coordinador. Si hay una trama de datos actual pendiente en el coordinador para el dispositivo que la solicita, el coordinador deberá enviar la trama al dispositivo usando uno de los mecanismos descritos en este apartado. Si no la hay para el dispositivo que la solicita, el coordinador deberá enviar una trama de datos sin solicitar confirmación y con la longitud del *payload* o carga a cero, indicando que no hay un dato presente, usando uno de los mecanismos descritos en este apartado.

La trama de datos que sigue a la confirmación del comando de solicitud de dato deberá ser transmitida usando uno de los siguientes mecanismos:

- **Sin usar acceso aleatorio ranurado**, si la subcapa MAC puede comenzar la transmisión de la trama de datos entre *aTurnaroundTime-RX-TX* y  $(aTurnaroundTime-RX-TX + aUnitBackoffPeriod)$  relojes ópticos, en un límite de slot de backoff, y que haya tiempo suficiente en el CAP para el mensaje, su apropiado *IFS* y la confirmación. Si la trama de confirmación no se recibe seguidamente a esta trama de datos, el proceso deberá comenzar de nuevo tras la recepción de un nuevo comando de solicitud de datos.
- Por otro lado, **usando acceso aleatorio ranurado**.

Si el dispositivo solicitante no recibe una trama de datos del coordinador dentro de *macMaxFrameTotalWait-Time* relojes ópticos del CAP en una VPAN con baliza habilitada, o en una VPAN sin baliza habilitada, o si el dispositivo solicitante recibe una trama de datos del coordinador con una longitud cero de *payload*, deberá concluir que no hay dato pendiente en el coordinador. Si el dispositivo solicitante recibe una trama de datos del coordinador, deberá enviar una trama de confirmación, si es requerida, para confirmar dicha recepción.

Si el subcampo de trama pendiente del campo de control de trama de la trama de datos recibida del coordinador está al valor uno, el dispositivo tiene aún más datos pendientes con el coordinador. En este caso debería extraer el dato enviando un nuevo comando de solicitud de datos al coordinador.

