

UNIVERSIDAD DE MÁLAGA
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
DE TELECOMUNICACIÓN



TESIS DOCTORAL

Modelado del retardo de transmisión en Bluetooth 2.0 + EDR

José Rafael Luque Giráldez

Málaga, 2010



AUTOR: José Rafael Luque Giráldez

Edita: Servicio de Publicaciones de la Universidad de Málaga

ISBN: 978-84-9747-590-7



Esta obra está sujeta a una licencia Creative Commons:

Reconocimiento - No comercial - SinObraDerivada (cc-by-nc-nd):

[Http://creativecommons.org/licenses/by-nc-nd/3.0/es](http://creativecommons.org/licenses/by-nc-nd/3.0/es)

Cualquier parte de esta obra se puede reproducir sin autorización
pero con el reconocimiento y atribución de los autores.

No se puede hacer uso comercial de la obra y no se puede alterar, transformar o hacer obras derivadas.

Esta obra se encuentra depositada en el Repositorio Institucional de la Universidad
de Málaga (RIUMA). <http://riuma.uma.es>

DR. EDUARDO CASILARI PÉREZ, PROFESOR TITULAR, Y DRA. MARÍA JOSÉ MORÓN FERNÁNDEZ,
INVESTIGADORA, DEL DEPARTAMENTO DE TECNOLOGÍA ELECTRÓNICA DE LA UNIVERSIDAD DE
MÁLAGA.

INFORMAN:

Que D. José Rafael Luque Giráldez, Ingeniero de Telecomunicación, ha realizado en el
Departamento de Tecnología Electrónica de la Universidad de Málaga, bajo su dirección el
trabajo de investigación correspondiente a su Tesis Doctoral titulada:

«MODELADO DEL RETARDO DE TRANSMISIÓN EN BLUETOOTH 2.0 + EDR»

Revisado el presente trabajo, estiman que puede ser presentado al Tribunal que ha de
juzgarlo. Por tanto AUTORIZAN y AVALAN la presentación de esta Tesis en la Universidad
de Málaga.

Málaga, a 23 de julio de 2010

Fdo. Eduardo Casilari Pérez
Profesor Titular del Dpto. de
Tecnología Electrónica

Fdo. María José Morón Fernández
Investigadora del Dpto. de
Tecnología Electrónica

Departamento de Tecnología Electrónica
E.T.S.I. de Telecomunicación
Universidad de Málaga

TESIS DOCTORAL

Modelado del retardo de transmisión en Bluetooth 2.0 + EDR

Autor:

José Rafael Luque Giráldez
Ingeniero de Telecomunicación

Directores:

D. Eduardo Casilari Pérez
Doctor Ingeniero de Telecomunicación

D^a. María José Morón Fernández
Ingeniera de Telecomunicación
Doctora por la Universidad de Málaga

A mi familia

A los padres de María José

*A María José, por demostrarme en el día a día
el valor de la fuerza que nace de la fe*

Agradecimientos

Quisiera expresar mis agradecimientos a todas las personas que han compartido conmigo la realización de esta tesis, especialmente:

A Eduardo Casilari Pérez por haber confiado en mí, por el tiempo dedicado, el interés mostrado y su gran implicación en la elaboración de esta tesis.

A María José por su apoyo incondicional, y por haber compartido y afrontado conmigo los retos y dificultades presentados durante la realización de esta tesis.

A Antonio Díaz Estrella por incentivar el desarrollo de la investigación, por sus sabios consejos y, sobre todo, por su ejemplaridad.

A Alfredo García Lopera por haber facilitado el analizador de espectro de Agilent Technologies.

A la Consejería de Educación de la Junta de Andalucía, por la concesión de una licencia retribuida por estudios, de un año de duración, para la finalización de esta tesis doctoral.

Finalmente, se ha de reconocer que esta tesis ha sido posible gracias a los proyectos que, parcialmente, han financiado su realización: «Proyecto AIRES: Aplicaciones Integradas en Redes Multiesfera (TIC2003-07953-C02-01)», «Proyecto ABRIL: Aplicaciones Biomédicas en Redes Inalámbricas Heterogéneas» (TEC2006-12211-C02-01), «Proyecto RAMAS: Redes de Arquitectura Mallada para Aplicaciones Sociosanitarias» (TEC2009-13763-C02-01).

Resumen

El objetivo principal de la presente tesis ha sido modelar el retardo de transmisión para el perfil de puerto serie SPP (*Serial Port Profile*) en Bluetooth 2.0 + EDR, tanto para el modo básico, compatible con Bluetooth 1.1, como para el modo EDR (*Enhanced Data Rate*). El estudio desarrollado ha consistido en formular expresiones matemáticas para calcular el retardo de transmisión, en los sentidos maestro-esclavo y esclavo-maestro.

La estimación teórica del retardo de transmisión se ha llevado a cabo, en primer lugar, asumiendo condiciones ideales, es decir, considerando que no se producen retransmisiones a nivel *Baseband*, para a continuación extender el modelo considerando el efecto de las posibles pérdidas en las transmisiones.

Posteriormente, la validación empírica del modelo desarrollado se ha llevado a cabo en una *piconet* Bluetooth utilizando la pila de protocolos BlueZ. En el caso de la evaluación del modelo sin considerar la ocurrencia de retransmisiones, se han empleado dispositivos reales con interfaz USB (*Universal Serial Bus*) de distintos fabricantes de gran implantación en productos Bluetooth comerciales: CSR (*Cambridge Silicon Radio*), ISS (*Integrated System Solution Corp.*) y *Broadcom Corporation*.

Para los dispositivos CSR se ha realizado un análisis exhaustivo, comparando el retardo previsto teóricamente con el obtenido empíricamente e introduciendo las correcciones necesarias para contemplar dos fenómenos que distorsionan el retardo medido, y enmascaran la validez del modelo formulado: el efecto de la capa de transporte y de la fragmentación según el tamaño de los *buffers* del dispositivo. En este sentido, a partir del estudio empírico de estos dos efectos, se ha podido concluir que:

1. El retardo asociado a la capa de transporte HCTL (*Host Controller Transport Layer*) USB resulta ser independiente del sentido de transmisión, y está presente tanto en dispositivos Bluetooth 2.0+ EDR con versión de *firmware* 0x7a6 y 0xc5c, como en dispositivos Bluetooth 2.1 + EDR.
2. El retardo introducido por la fragmentación realizada de acuerdo con el tamaño de los *buffers* del dispositivo es un efecto que tiene lugar tanto en el modo básico como en el modo EDR, para el sentido de transmisión maestro-esclavo, y en dispositivos Bluetooth 2.0 + EDR, a partir de la versión de *firmware* 0xc5c.

Así, para evaluar el efecto sobre el retardo de la transferencia USB y, por ende, constatar la validez del modelo, ha sido necesario caracterizar la capa de transporte HCTL. Como resultado de este análisis se ha obtenido empíricamente una expresión matemática que aproxima el retardo registrado introducido por la capa de transporte, que ha sido integrada en el modelo original con resultados satisfactorios.

Sin embargo, para los dispositivos ISS y *Broadcom Corporation*, el análisis no ha podido ser tan detallado, debido a que no se disponen de las mismas herramientas que en el caso de CSR y a que se ha comprobado que los propios dispositivos no funcionan correctamente en todos los casos.

Una vez ratificada la utilidad del modelo asumiendo condiciones ideales (sin retransmisiones), teniendo en cuenta el hecho de que Bluetooth opera en la banda ISM, compartida con otras tecnologías inalámbricas (tales como ZigBee o Wi-Fi), se ha considerado que los escenarios sin pérdidas resultan ser poco realistas para la mayoría de las aplicaciones. Por consiguiente, se ha completado el modelo con el fin de contemplar el efecto de las retransmisiones en el cálculo del retardo asociado al perfil SPP. La extensión se ha llevado a cabo para el modo de operación básico (modo por defecto) del protocolo L2CAP (*Logical Link Control and Adaptation Protocol*), en el que no se realizan retransmisiones a nivel L2CAP y que es el que generalmente se utiliza en las aplicaciones de Bluetooth, debido a que garantiza la interoperatividad con versiones anteriores a Bluetooth 2.0 + EDR. Así pues, el objetivo perseguido ha sido incorporar el efecto de las retransmisiones a nivel *Baseband* en el cálculo del retardo.

Tanto la formulación del modelo como su validación, realizada empíricamente, se han llevado a cabo determinando el retardo de transmisión a partir de la tasa de error de bit

(BER, *Bit Error Rate*), para cada una de las modulaciones Bluetooth, en función de la relación señal/ruido, SNR (*Signal-to-Noise Ratio*), asumiendo para ello que el canal de transmisión Bluetooth se puede modelar como un canal de ruido blanco gaussiano (AWGN, *Additive White Gaussian Noise*), es decir, considerando que el efecto del ruido es equivalente a la adición de ruido gaussiano de potencia constante en la banda considerada.

Para la validación del modelo considerando el efecto de las retransmisiones, se han empleado dispositivos del fabricante CSR y se ha diseñado un escenario que permite reproducir empíricamente el modelo de canal gaussiano propuesto teóricamente, utilizando un generador de ruido y atenuadores. De esta forma se consigue que las transmisiones entre dos dispositivos Bluetooth se vean afectadas por un factor de atenuación y la adición de un nivel de ruido blanco gaussiano. En las condiciones indicadas, para diferentes valores de relación SNR se ha medido la tasa BER, comparándola con la prevista teóricamente a partir de la relación SNR impuesta. La comparación del retardo medido con el retardo calculado analíticamente a partir de la tasa BER medida en distintos casos, ha permitido demostrar el buen ajuste del modelo propuesto a la hora de predecir el retardo de las transmisiones a partir de la relación SNR existente.

Abstract

The main purpose of the present thesis has been to model the transmission delay for the Serial Port Profile (SPP) within the scope of Bluetooth 2.0 + EDR, for both the basic mode, compatible with Bluetooth 1.1, and the Enhanced Data Rate (EDR) mode. The study consists of the development of the mathematical expressions to compute the transmission delay for master-slave and slave-master communications.

In a first stage, the theoretical estimation of the transmission delay has been carried out assuming ideal conditions, i.e., retransmissions at Baseband level have not been considered to occur. Subsequently the proposed model has been extended by taking into account the possible effect of the occurrence of retransmissions.

In a second stage, the model has been empirically validated in a real Bluetooth piconet by using the BlueZ protocol stack. For the evaluation of the model without packet retransmissions, currently available Bluetooth USB devices from different vendors, widely extended in commercial Bluetooth products, have been employed: CSR (Cambridge Silicon Radio), ISS (Integrated System Solution Corp.) and Broadcom Corporation.

In the case of CSR devices an exhaustive analysis has been carried out by comparing the theoretically predicted delay by the model with the measured one. Moreover several improvements have been introduced in order to account for two physical effects that distort the measured delay and disguise the validity of the proposed model: the effect of the transport layer and the device buffer size. In this sense, from the empirical study of these effects, we can conclude that:

1. The delay introduced by the USB (Universal Serial Bus) HCTL (Host Controller Transport Layer) transport layer does not depend on the transmission direction. Besides, it has been detected in Bluetooth 2.0 + EDR devices with firmware versions 0x7a6 and 0xc5c, and also in Bluetooth 2.1 + EDR devices.
2. The delay added by the fragmentation caused by device buffer size has been observed for firmware version 0xc5c (and newer versions), for both the basic mode and the EDR mode, and only in master-slave transmission direction.

So that, in order to evaluate the impact of the time consumed by the USB transfers, and therefore, to establish the validity of the proposed model, the dynamics of the HCTL transport layer have been studied and characterized. As a result of this analysis a mathematical expression (empirically derived from the measurements) that closely approximates the delay introduced by the transport layer has been obtained. This equation has been successfully integrated in the original model for the computation of the transmission delay.

On the other hand, for ISS and Broadcom Corporation devices it has not been possible, due to the lack of appropriate tools, to perform an analysis as detailed as for the CSR case and also due to the fact that the devices themselves do not always work properly.

Once the model has proved to be useful under ideal conditions (no retransmissions), it has been considered that the assumption of no loss packets may be unrealistic in most practical scenarios, taking into account that Bluetooth operates in the unlicensed ISM band, which is shared among several wireless technologies (such as ZigBee or Wi-Fi). Therefore, the study has been extended in order to cope with the effect of retransmissions in the computation of the SPP associated delay. The extension of the model has been carried out to include the basic mode of operation, which is the default mode, of the L2CAP (Logical Link Control and Adaptation Protocol) protocol. In this mode of operation no retransmissions at L2CAP level occurs. Furthermore, this mode is massively employed by Bluetooth applications as it ensures the interworking with versions older than Bluetooth 2.0 + EDR. So, the pursued target has been to include the effect of Baseband retransmissions in the delay computation.

The formulation of the model and its empirically validation have been carried out by calculating the transmission delay from the Bit Error Rate (BER) for each of the Bluetooth modulations as a function of the Signal-to-Noise Ratio (SNR). Making the assumption that

the Bluetooth transmission channel could be characterized as an Additive White Gaussian Noise (AWGN) channel, that is, by considering that the effect of the noise is equivalent to the addition of constant power Gaussian noise over the considered bandwidth.

In order to evaluate the suggested model without neglecting the occurrence of retransmissions, CSR devices have been employed again, and a scenario that allows to reproduce empirically the theoretical AGWN has been designed by using a noise generator and signal attenuators. Thus, Bluetooth transmissions between the test devices are affected by an attenuation factor and the addition of a certain level of white Gaussian noise. In this testbed the BER has been measured for different values of the noise level set. Additionally, this estimated BER has been compared against the BER theoretically predicted for that given value of SNR. The comparison of the measured delay with the analytically computed delay from the measured BER in different cases, has proved the fair adjustment of the proposed model when predicting the transmission delay from the existing SNR.

Índice general

Índice de figuras	VII
Índice de tablas	XIX
Lista de acrónimos, abreviaturas y símbolos	XXI
1. Introducción	1
2. Fundamentos teóricos de Bluetooth	7
2.1. Bluetooth 1.1	8
2.1.1. Nivel RF (<i>Radio Frequency</i>)	10
2.1.2. Nivel BB (<i>Baseband</i>)	11
2.1.2.1. Canal físico	12
2.1.2.2. Enlaces físicos	13
2.1.2.3. Estados y modos de operación	14
2.1.2.4. Paquetes Bluetooth	18
2.1.2.5. Técnicas de corrección de errores empleadas	22
2.1.3. Nivel LM (<i>Link Manager</i>)	23
2.1.4. Nivel HCI (<i>Host Controller Interface</i>)	24
2.1.5. Nivel L2CAP (<i>Logical Link Control and Adaptation Protocol</i>)	25
2.2. Bluetooth 1.2, 2.0 + EDR y 2.1 + EDR	25
2.2.1. Arquitectura de protocolos Bluetooth 1.2	27
2.2.1.1. Nivel RF	29

2.2.1.2. Nivel BB	29
2.2.1.3. Nivel LM	30
2.2.1.4. Nivel L2CAP	30
2.2.2. Arquitectura del transporte de datos	36
2.2.2.1. Canales físicos	37
2.2.2.2. Enlaces físicos	38
2.2.2.3. Transportes lógicos	38
2.2.2.4. Enlaces lógicos	40
2.2.2.5. Canales L2CAP	41
2.2.3. Paquetes Bluetooth a partir de Bluetooth 1.2	41
2.2.3.1. Tipos de paquetes	41
2.2.3.2. Estructura del paquete	42
2.3. Perfiles Bluetooth	46
2.3.1. Perfil SPP y protocolo RFCOMM	46
3. Estado de la técnica	51
3.1. Estudios de rendimiento relacionados con los esquemas de transmisión TDD	53
3.2. Estudios de rendimiento relacionados con procedimientos de transmisión . .	55
3.3. Estudios de rendimiento relacionados con la coexistencia IEEE 802.11-Bluetooth	65
3.4. Relación entre la probabilidad de error y la relación SNR para las modulaciones Bluetooth	73
3.4.1. Diseño y fabricación de moduladores y demoduladores	74
3.4.2. Estudios teóricos genéricos	75
3.4.3. Estudios aplicados a Bluetooth	78
3.4.4. Estudios específicos de Bluetooth	80
3.5. Resumen y conclusiones	81
4. Cálculo del retardo de transmisión para el perfil SPP incluyendo el modo EDR	87
4.1. Introducción	87
4.2. Modelo para el perfil SPP en Bluetooth 1.1 en condiciones ideales (sin retransmisiones)	89
4.3. Adaptación del modelo a Bluetooth 2.0 + EDR en condiciones ideales	94

4.4. Adaptación del modelo a Bluetooth 2.0 + EDR considerando el efecto de las retransmisiones	104
4.4.1. Caracterización de las modulaciones digitales empleadas en Bluetooth 2.0 + EDR en un canal AWGN	106
4.4.1.1. Modulación GFSK	108
4.4.1.2. Modulación $\pi/4$ DQPSK	109
4.4.1.3. Modulación 8 DPSK	111
4.4.1.4. Comparación del rendimiento de las modulaciones GFSK, $\pi/4$ DQPSK y 8 DPSK	113
4.4.2. Cálculo de la tasa PER	114
4.4.3. Cálculo del retardo de transmisión	118
4.4.4. Análisis de la aplicación del modelo	122
5. Evaluación empírica del modelo teórico sin retransmisiones	131
5.1. Introducción	131
5.2. Descripción del sistema de pruebas	132
5.2.1. Sistema operativo y dispositivos Bluetooth empleados	132
5.2.2. Procedimiento de medida	133
5.3. Dispositivos CSR	136
5.3.1. Evaluación inicial: Análisis del efecto del parámetro ACL MTU	138
5.3.2. Análisis del efecto de la configuración de servicio	150
5.3.2.1. Sentido de transmisión maestro-esclavo	152
5.3.2.2. Sentido de transmisión esclavo-maestro	156
5.3.3. Caracterización de la capa de transporte	160
5.3.4. Modificación del modelo teórico para contemplar el efecto de la capa de transporte	162
5.3.4.1. Paquetes DH en el modo EDR	165
5.3.4.2. Paquetes DM y DH en el modo básico	169
5.3.5. Modificación del modelo teórico para contemplar el efecto del transporte USB y la fragmentación realizada al valor del parámetro ACL MTU en el sentido de transmisión maestro-esclavo	173
5.3.5.1. Cálculo de las componentes básicas del retardo de transmisión teniendo en cuenta el efecto del parámetro ACL MTU	173

5.3.5.2.	Cálculo de las componentes básicas del retardo de transmisión teniendo en cuenta el efecto del parámetro ACL MTU y de la transferencia USB	175
5.3.5.3.	Cálculo del retardo a nivel de RFCOMM	180
5.3.5.4.	Presentación de los resultados	184
5.3.6.	Modificación del modelo teórico para contemplar el efecto de la capa de transporte para tamaños de datos de usuario superiores a 1008 octetos	196
5.3.6.1.	Paquetes DM y DH en el modo básico	198
5.3.6.2.	Paquetes DH en el modo EDR	205
5.3.7.	Recapitulación y conclusiones	213
5.4.	Dispositivos ISS	221
5.5.	Dispositivos <i>Broadcom</i>	229
6.	Evaluación empírica del modelo teórico teniendo en cuenta el efecto de las retransmisiones	233
6.1.	Metodología de medida	233
6.2.	Análisis de los resultados	237
6.2.1.	Medición de la tasa BER	238
6.2.2.	Comparación de la tasa BER medida con la obtenida teóricamente a partir de la relación SNR	243
6.2.2.1.	Medida de la potencia real de ruido	243
6.2.2.2.	Medida de la potencia real de señal	247
6.2.2.3.	Cálculo de la relación SNR a la entrada del receptor	254
6.2.2.4.	BER obtenida a partir de la relación SNR medida	255
6.2.3.	Análisis del efecto de las retransmisiones	256
7.	Conclusiones y líneas futuras	273
7.1.	Conclusiones	273
7.1.1.	Conclusiones sobre el modelo teórico sin retransmisiones	274
7.1.1.1.	Dispositivos CSR	276
7.1.1.2.	Dispositivos ISS	279
7.1.1.3.	Dispositivos <i>Broadcom</i>	280

7.1.2. Conclusiones sobre el modelo teórico con retransmisiones	281
7.1.2.1. Conclusiones derivadas del modelo teórico	282
7.1.2.2. Conclusiones sobre la validación empírica del modelo con retransmisiones	283
7.2. Líneas futuras	286
7.3. Publicaciones derivadas del desarrollo de la tesis	287
7.3.1. Revistas incluidas en el JCR	287
7.3.2. Congresos Internacionales	287
7.3.3. Otras publicaciones	288
A. Medidas realizadas con el S.O. Linux Ubuntu 9.10 Karmic Koala	291
Bibliografía	297

Índice de figuras

2.1. Arquitectura de protocolos Bluetooth 1.1. (adaptada de: [Bluetooth SIG01, parte H:1])	10
2.2. Esquema de transmisión de paquetes <i>multi-slot</i> (fuente: [Bluetooth SIG01, parte B])	13
2.3. Diagrama de estados definido en Bluetooth 1.1 (fuente: [Bluetooth SIG01, parte B])	15
2.4. Estructura del paquete Bluetooth 1.1; Formatos de las cabeceras del paquete y del <i>payload</i> . (fuente: [Bluetooth SIG01, parte B])	20
2.5. Arquitectura de protocolos Bluetooth (fuente: [Bluetooth SIG03, Bluetooth SIG04, Bluetooth SIG07, vol. 1, parte A])	28
2.6. Modificación de los estados del nivel BB a partir de la versión Bluetooth 1.2 (fuente: [Bluetooth SIG03, Bluetooth SIG04, Bluetooth SIG07, vol. 2, parte B])	30
2.7. Entidades del nivel L2CAP definidas a partir de Bluetooth 1.2 (fuente: [Bluetooth SIG03, Bluetooth SIG04, Bluetooth SIG07, vol. 3, parte A]).	31
2.8. Arquitectura del transporte de datos Bluetooth (fuente: [Bluetooth SIG03, Bluetooth SIG04, Bluetooth SIG07, vol. 1, parte A])	37
2.9. Estructura del paquete Bluetooth 2.0 + EDR y 2.1 + EDR. Formatos de las cabeceras del paquete y del <i>payload</i> . Fuente: [Bluetooth SIG04, Bluetooth SIG07]	45
2.10. Estructura de la trama del Modo de Operación Básico del protocolo TS 07.10 (fuente: [Bluetooth SIG01, parte F:1])	47
4.1. Arquitectura de protocolos Bluetooth (adaptado de: [Moron08])	88

4.2. Trama de Información en el modo básico de L2CAP (<i>B-Frame</i>) (fuente: [Bluetooth SIG07, vol. 3, parte A]).	90
4.3. Fragmentación realizada en el nivel RFCOMM cuando el tamaño de los datos de usuario excede el valor de L_R , dado por la ecuación 4.4 (fuente: [Moron08]).	91
4.4. Transmisión de una trama de L2CAP cuando se excede la capacidad de un paquete de 5 <i>slots</i>	91
4.5. Esquema de utilización de paquetes BB según se incrementa el tamaño de los datos de usuario	97
4.6. Representación del número de <i>time slots</i> ocupados y de la modulación empleada en la transmisión de x octetos de usuario, según el esquema de utilización de paquetes BB para Bluetooth 2.0 + EDR (figura 4.5).	98
4.7. Tiempos introducidos en un paquete Bluetooth al incorporar el modo EDR. .	99
4.8. Retardo medio calculado, en el sentido maestro-esclavo, para el perfil SPP empleando paquetes DH y considerando un intervalo de <i>poll</i> de 25 ms (40 <i>slots</i>).102	
4.9. Retardo medio calculado, en el sentido maestro-esclavo, para el perfil SPP empleando paquetes DM y considerando un intervalo de <i>poll</i> de 25 ms (40 <i>slots</i>).104	
4.10. Gráfica teórica de la probabilidad de error de bit, P_b , equivalente a la tasa BER, para las modulaciones FSK (con detección coherente y no coherente), y GFSK con índices de modulación $h = 0,28$ y $h = 0,35$, en función de la relación SNR por bit, $\gamma_b = \frac{E_b}{N_0}$	110
4.11. Gráfica teórica de la probabilidad de error de bit, P_b , equivalente a la tasa BER, para la modulación $\pi/4$ DQPSK en función de la relación SNR por bit γ_b , contrastando la fórmula general 4.35 con la fórmula 4.30 (función Q de Marcum) particularizada con los parámetros dados por 4.36.	112
4.12. Gráfica teórica de la probabilidad de error de bit, P_b , equivalente a la tasa BER, para la modulación 8 DQPSK en función de la relación SNR por bit, γ_b , contrastando la fórmula general 4.35 con la cota 4.37, y la aproximación 4.41, equivalente a 4.40.	114
4.13. Gráfica teórica de la probabilidad de error de bit, P_b , equivalente a la tasa BER, para las modulaciones GFSK (con índice de modulación $h = 0,35$), $\pi/4$ DQPSK y 8 DPSK, en función de la relación SNR por bit, γ_b	115

4.14. Tiempo transcurrido desde que se inicia la transmisión de un paquete de 5 slots hasta que se realiza la retransmisión, particularizando $t_{RTx}(N)$ al valor máximo, $t_{RTx_{max}}(N)$ (fuente: [Moron08])	120
4.15. Tiempo transcurrido desde que se inicia la transmisión de un paquete de 5 slots hasta que se realiza la retransmisión, particularizando $t_{RTx}(N)$ al valor mínimo, $t_{RTx_{min}}(N)$ (fuente: [Moron08])	121
4.16. Retardo de transmisión $\overline{t_{TX_l}}(M(N))$, en función de la relación SNR por bit, γ_b , utilizando paquetes del modo básico y EDR	126
4.17. Efecto de la relación SNR en las transmisiones maestro-esclavo para el perfil SPP	128
4.18. Efecto de la relación SNR en las transmisiones maestro-esclavo para el perfil SPP, utilizando la política de selección de paquetes EDR reflejada en la figura 4.5.	129
5.1. Esquema general de pruebas empleado para la caracterización del retardo de transmisión Bluetooth, utilizado para evitar las interferencias	134
5.2. Efecto de la configuración del parámetro ACL MTU para la versión del <i>firmware</i> 0xc5c	139
5.3. Comparación del retardo medio de transmisión maestro-esclavo previsto teóricamente con el medido para un valor del parámetro ACL MTU de 1021 y de 512 octetos (Revisión de HCI: 0xc5c).	140
5.4. Comparación del retardo medio de transmisión maestro-esclavo previsto teóricamente con el medido para un valor del parámetro ACL MTU de 384 y de 310 octetos (Revisión de HCI: 0xc5c).	141
5.5. Comparación de los retardos medios de transmisión obtenidos para las versiones de <i>firmware</i> 0x7a6 y 0xc5c, con un valor del parámetro ACL MTU de 384 octetos, en ambos casos.	142
5.6. Comparación del retardo medio de transmisión maestro-esclavo previsto teóricamente con el retardo medido para la versión de <i>firmware</i> 0x7a6, con un valor del parámetro ACL MTU de 384 octetos.	143
5.7. Comparación de los retardos medios de transmisión medidos y de su desviación típica para las versiones de <i>firmware</i> 0x7a6 y 0xc5c, con valores del parámetro ACL MTU de 384 octetos y 1021 octetos, respectivamente. . . .	144

5.8. Comparación de los retardos obtenidos en el sentido de transmisión maestro-esclavo con dispositivos Bluetooth 2.0 + EDR y Bluetooth 2.1 + EDR para un valor del parámetro ACL MTU de 1021 octetos.	145
5.9. Comparación de los retardos obtenidos en el sentido de transmisión maestro-esclavo con dispositivos Bluetooth 2.0 + EDR y Bluetooth 2.1 + EDR.	146
5.10. Comparación de los retardos medios de transmisión obtenidos para la versión de <i>firmware</i> 0xc5c, utilizando paquetes DM, cuando el parámetro ACL MTU toma los valores de 200, 310, 512 y 680 octetos	147
5.11. Comparación del retardo medio previsto teóricamente y del retardo medio de transmisión obtenido con dispositivos Bluetooth 1.1 y Bluetooth 2.0 + EDR (Revisión de HCI 0xc5c)	148
5.12. Comparación del retardo medio previsto teóricamente y del retardo medio de transmisión obtenido con dispositivos Bluetooth 2.0 + EDR (Revisión de HCI 0x7a6)	149
5.13. Comparación de los retardos medios de transmisión obtenidos para la versión de <i>firmware</i> 0xc5c, utilizando paquetes DH del modo básico, cuando el parámetro ACL MTU toma los valores de 256, 310, 384, 512 y 680 octetos . .	150
5.14. Efecto de la configuración del parámetro ACL MTU para la versión del <i>firmware</i> 0xc5c en el sentido de transmisión esclavo-maestro	151
5.15. Efecto de la configuración de la calidad de servicio, en el sentido de transmisión maestro-esclavo, para la versión de <i>firmware</i> 0x7a6 y para un valor del parámetro ACL MTU de 384 octetos	153
5.16. Efecto de la configuración de la calidad de servicio, en el sentido de transmisión maestro-esclavo, para la versión de <i>firmware</i> 0xc5c para paquetes DH y valores del parámetro ACL MTU de 1021 y 512 octetos.	154
5.17. Efecto de la configuración de la calidad de servicio, en el sentido de transmisión maestro-esclavo, para la versión de <i>firmware</i> 0xc5c para paquetes DH y valores del parámetro ACL MTU de 384 y 310 octetos.	155
5.18. Efecto de la configuración de la calidad de servicio, en el sentido de transmisión maestro-esclavo, para la versión de <i>firmware</i> 0xc5c para paquetes DM.	156

5.19.Efecto de la configuración de la calidad de servicio, en el sentido de transmisión esclavo-maestro, para la versión de <i>firmware</i> 0x7a6 y para un valor del parámetro ACL MTU de 384 octetos	158
5.20.Efecto de la configuración de la calidad de servicio, en el sentido de transmisión esclavo-maestro, para la versión de <i>firmware</i> 0xc5c	159
5.21.Caracterización de la capa de transporte utilizando el modo <i>loopback</i> local .	161
5.22.Ajuste lineal con mínimos cuadrados del retardo asociado a la capa de transporte HCTL	163
5.23.Comparación del retardo medido en el modo <i>loopback</i> local para valores del valor del parámetro ACL MTU de 1021 y 420 octetos.	164
5.24.Comparación del retardo medido y el teórico incluyendo el retardo introducido por la capa de transporte USB, en el sentido maestro-esclavo, para un valor del parámetro ACL MTU de 1021 octetos (Revisión de HCI: 0xc5c)	166
5.25.Comparación del retardo medido y el teórico incluyendo el retardo introducido por la capa de transporte USB, en el sentido esclavo-maestro, para paquetes DH y un valor del parámetro ACL MTU de 1021 octetos, con corrección para longitudes superiores a $M_R = 1013$ octetos (Revisión de HCI: 0xc5c)	168
5.26.Comparación del retardo medido y el teórico incluyendo el retardo introducido por la capa de transporte USB, en el sentido maestro-esclavo, con corrección para longitudes superiores a $M_R = 1013$ octetos, para un valor del parámetro ACL MTU de 384 octetos (Revisión de HCI: 0x7a6)	169
5.27.Comparación del retardo medido y el teórico, incluyendo el retardo introducido por la capa de transporte USB, en el sentido maestro-esclavo, para paquetes DH y DM (Revisión de HCI: 0xc5c)	171
5.28.Efecto de la fragmentación al tamaño del parámetro ACL MTU y de la capa de transporte para la transferencia de una trama L2CAP que encapsula una trama de RFCOMM con 678 octetos de datos de usuario, respectivamente. . .	176
5.29.Efecto de la fragmentación al tamaño del parámetro ACL MTU y de la capa de transporte para la transferencia de una trama L2CAP que encapsula una trama de RFCOMM de longitud máxima (1008 octetos de datos de usuario).	185

5.30.Comparación del retardo medido y el teórico incluyendo la suma de los efectos de la fragmentación según el parámetro ACL MTU y de la capa de transporte USB, en el sentido maestro-esclavo, para paquetes DH del modo EDR y valores del parámetro ACL MTU de 680 y 512 octetos (Revisión de HCI: 0xc5c).	187
5.31.Comparación del retardo medido y el teórico incluyendo la suma de los efectos de la fragmentación según el parámetro ACL MTU y de la capa de transporte USB, en el sentido maestro-esclavo, para paquetes DH del modo EDR y valores del parámetro ACL MTU de 420 y 384 octetos (Revisión de HCI: 0xc5c)	188
5.32.Comparación del retardo medido y el teórico incluyendo la suma de los efectos de la fragmentación según el parámetro ACL MTU y de la capa de transporte USB, en el sentido maestro-esclavo, para paquetes DH del modo EDR y valores del parámetro ACL MTU de 350 y 310 octetos (Revisión de HCI: 0xc5c)	189
5.33.Comparación del retardo medido y el teórico incluyendo la suma de los efectos de la fragmentación según el parámetro ACL MTU y de la capa de transporte USB, en el sentido maestro-esclavo, para paquetes DH del modo básico y valores del parámetro ACL MTU de 680 y 384 octetos (Revisión de HCI: 0xc5c)	192
5.34.Comparación del retardo medido y el teórico incluyendo la suma de los efectos de la fragmentación según el parámetro ACL MTU y de la capa de transporte USB, en el sentido maestro-esclavo, para paquetes DH del modo básico y valores del parámetro ACL MTU de 310 y 256 octetos (Revisión de HCI: 0xc5c)	193
5.35.Comparación del retardo medido y el teórico incluyendo la suma de los efectos de la fragmentación según el parámetro ACL MTU y de la capa de transporte USB, en el sentido maestro-esclavo, para paquetes DM y valores del parámetro ACL MTU de 680 y 200 octetos (Revisión de HCI: 0xc5c)	194

5.36.Comparación del retardo medido y el teórico incluyendo la suma de los efectos de la fragmentación según el parámetro ACL MTU y de la capa de transporte USB, en el sentido maestro-esclavo, para paquetes DH y DM del modo básico y un valor del parámetro ACL MTU de 512 octetos (Revisión de HCI: 0xc5c)	195
5.37.Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DH básicos: DH1, DH3 y DH5 y valores del parámetro ACL MTU de 680 y 384 octetos (Revisión de HCI: 0xc5c).	199
5.38.Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DH básicos: DH1, DH3 y DH5 y valores del parámetro ACL MTU de 310 y 256 octetos (Revisión de HCI: 0xc5c)	200
5.39.Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DM y y valores del parámetro ACL MTU de 680 y 200 octetos (Revisión de HCI: 0xc5c).	201
5.40.Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DM y DH básicos, y un valor del parámetro ACL MTU de 512 octetos (Revisión de HCI: 0xc5c).	202
5.41.Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DM (Revisión de HCI: 0x7a6).	203
5.42.Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido esclavo-maestro, para paquetes DH y DM del modo básico (Revisión de HCI: 0xc5c).	204
5.43.Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DH y un valor del parámetro ACL MTU de 1021 octetos (Revisión de HCI: 0xc5c)	205
5.44.Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DH y valores del parámetro ACL MTU de 740 y 680 octetos (Revisión de HCI: 0xc5c)	207

5.45.Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DH y valores del parámetro ACL MTU de 512 y 420 octetos (Revisión de HCI: 0xc5c)	208
5.46.Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DH y valores del parámetro ACL MTU de 384 y 350 octetos (Revisión de HCI: 0xc5c)	209
5.47.Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DH y valores del parámetro ACL MTU de 310 y 256 octetos (Revisión de HCI: 0xc5c)	210
5.48.Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18 con la corrección dada por 5.18, en el sentido maestro-esclavo, para paquetes DH y y valores del parámetro ACL MTU de 1021 y 420 octetos (Revisión de HCI: 0xc5c)	211
5.49.Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18 con la corrección dada por 5.18, en el sentido maestro-esclavo, para paquetes DH y valores del parámetro ACL MTU de 384 y 310 octetos (Revisión de HCI: 0xc5c)	212
5.50.Efecto de la configuración de la calidad de servicio en dispositivos ISS, para paquetes DH y un valor del parámetro ACL MTU de 1021 octetos.	222
5.51.Comparación del retardo medio de transmisión teórico a partir de la ecuaciones 4.1 y 5.37 con el medido para dispositivos ISS en el sentido maestro-esclavo.	225
5.52.Diferencia entre el retardo medido y el retardo calculado mediante la ecuación 5.37	226
5.53.Comparación entre el retardo medido y el retardo teórico calculado a partir de la ecuación 5.37, teniendo en cuenta la estimación del retardo introducido por la capa de transporte HCTL.	227
5.54.Retardos de transmisión obtenidos con dispositivos del fabricante ISS en el sentido de transmisión maestro-esclavo, empleando paquetes DM.	228
5.55.Representación del retardo medio de transmisión sin configurar calidad de servicio, medido con dispositivos <i>Broadcom</i> , para paquetes DH y valor del parámetro ACL MTU de 1017 octetos.	229

5.56. Efecto de la configuración de la calidad de servicio en dispositivos <i>Broadcom</i> , para paquetes DM y un valor del parámetro ACL MTU de 1017 octetos.	231
5.57. Comparación del retardo medio de transmisión teórico con el medido, utilizando paquetes DM, para dispositivos <i>Broadcom</i>	232
6.1. Diagrama de conexión de componentes	234
6.2. Vista general del sistema de medida	234
6.3. Placas Bluetooth de desarrollo BlueGiga WT11-E	235
6.4. Instrumentación utilizada para modelar el comportamiento de un canal gaussiano	238
6.5. Tasa BER obtenida, a partir de la ecuación 6.1, para un nivel de atenuación de 45 dB y 4 dB de ruido en los sentidos de transmisión maestro-esclavo y esclavo-maestro.	240
6.6. Aplicación <i>BlueTest</i> del fabricante CSR	241
6.7. Analizador de espectro Agilent Technologies N9010A.	244
6.8. Capturas de la vista del analizador de espectro en la que se muestra las potencias de ruido medidas para el nivel de atenuación de 4 dB y para un ancho de banda de 2 MHz y de 80 MHz, en torno a la frecuencia central de 2441 MHz.	246
6.9. Captura del analizador de espectro en la que se muestra el espectro de potencia correspondiente a la transmisión de un paquete DH5 con 339 octetos, medido a la salida del combinador, a la frecuencia central de 2441 MHz, para el nivel de atenuación de 45 dB y un ancho de banda de 2 MHz.	249
6.10. Capturas del analizador de espectro en las que se muestra la variación de la potencia durante la transmisión de un paquete DH5 con 339 octetos, obtenidas a la salida del transmisor y del combinador	251
6.11. Capturas del analizador de espectro en las que se muestra la variación de la potencia durante la transmisión de un paquete 2-DH5 y 3-DH5 con 679 y 1021 octetos	252

6.12. Capturas del analizador de espectro en las que se muestran los espectros de potencia de las transmisiones de un paquete 2-DH5 y 3-DH5 con 679 y 1021 octetos, respectivamente, medidos a la salida del combinador, a la frecuencia central de 2441 MHz, para el nivel de atenuación de 45 dB y un ancho de banda de 2 MHz.	253
6.13. Comparación entre la curva teórica de la probabilidad de error de bit, P_b , para la modulación GFSK, y la tasa BER medida empíricamente para el modo básico de Bluetooth, en función de la relación SNR por bit, γ_b	256
6.14. Comparación del retardo medido con una atenuación de ruido de 8 dB (SNR=14,95 dB) y en ausencia de ruido, en el sentido de transmisión maestro-esclavo, con paquetes DM y un intervalo de $poll$ de 10 ms.	257
6.15. Comparación entre la curva teórica de la probabilidad de error de bit, P_b , para las modulaciones EDR, y la tasa BER medida empíricamente para el modo básico de Bluetooth, en función de la relación SNR por bit, γ_b	258
6.16. Efecto de la configuración de la calidad de servicio (T_{poll}), en ambos sentidos de transmisión, para una atenuación de ruido de 6 dB.	260
6.17. Ejemplo que muestra la corrección introducida en el modelo para contemplar, en el sentido de transmisión esclavo-maestro, el salto incremental que se produce tras el llenado del primer paquete de 5 slots (fuente: [Moron08]) . .	262
6.18. Tiempo transcurrido desde que se inicia la transmisión de un paquete de 5 slots hasta que se realiza la retransmisión, particularizando $t_{RTx}(N)$ al valor mínimo, $t_{RTx_{min}}(N)$	263
6.19. Comparación del retardo medido y de las curvas teóricas calculadas a partir de la tasa BER medida y de las tasas BER teóricas obtenidas a partir de la relación SNR mediante la ecuación 4.30 para los índices de modulación $h = 0,30$ y $h = 0,35$, en el sentido maestro-esclavo, utilizando paquetes DM y para un nivel de atenuación de ruido de 4 dB (SNR = 10,95 dB).	265
6.20. Comparación del retardo medido y de las curvas teóricas calculadas a partir de la tasa BER medida y de las tasas BER teóricas obtenidas a partir de la relación SNR mediante la ecuación 4.30 para los índices de modulación $h = 0,30$ y $h = 0,35$, en el sentido maestro-esclavo, utilizando paquetes DM y para niveles de atenuación de ruido de 5 y 6 dB (SNR = 11,95 dB y 12,95 dB). .	266

6.21. Comparación del retardo medido y de las curvas teóricas calculadas a partir de la tasa BER medida y de las tasas BER teóricas obtenidas a partir de la relación SNR mediante la ecuación 4.30 para los índices de modulación $h = 0,30$ y $h = 0,35$, en el sentido maestro-esclavo, utilizando paquetes DM y para niveles de atenuación de ruido de 7 y 8 dB (SNR = 13,95 dB y 14,95 dB).	267
6.22. Comparación del retardo medido y de las curvas teóricas calculadas a partir de la tasa BER medida y de las tasas BER teóricas obtenidas a partir de la relación SNR mediante la ecuación 4.30 para los índices de modulación $h = 0,30$ y $h = 0,35$, en el sentido esclavo-maestro, utilizando paquetes DM y para niveles de atenuación de ruido de 4 y 5 dB (SNR = 10,95 dB y 11,95 dB).	268
6.23. Comparación del retardo medido y de las curvas teóricas calculadas a partir de la tasa BER medida y de las tasas BER teóricas obtenidas a partir de la relación SNR mediante la ecuación 4.30 para los índices de modulación $h = 0,30$ y $h = 0,35$, en el sentido esclavo-maestro, utilizando paquetes DM y para niveles de atenuación de ruido de 5 y 6 dB (SNR = 11,95 dB y 12,95 dB).	269
6.24. Comparación del retardo medido y de las curvas teóricas calculadas a partir de la tasa BER medida y de las tasas BER teóricas obtenidas a partir de la relación SNR mediante la ecuación 4.30 para los índices de modulación $h = 0,30$ y $h = 0,35$, en el sentido esclavo-maestro, utilizando paquetes DM y para un nivel de atenuación de ruido de 6 dB (SNR = 12,95 dB).	270
6.25. Comparación del retardo medido y de las curvas teóricas calculadas a partir de la tasa BER medida y de las tasas BER teóricas obtenidas a partir de la relación SNR mediante la ecuación 4.30 para los índices de modulación $h = 0,30$ y $h = 0,35$, en el sentido esclavo-maestro, utilizando paquetes DM y para niveles de atenuación de ruido de 7 y 8 dB (SNR = 13,95 dB y 14,95 dB).	271
6.26. Comparación del retardo medido en el modo EDR y de la curva teórica calculada para paquetes DM a partir de la tasa BER medida, en el sentido maestro-esclavo, para un nivel de atenuación de ruido de 8 dB y $T_{poll}=10$ ms.	272
7.1. Esquema de utilización de paquetes BB según se incrementa el tamaño de los datos de usuario	275

A.1. Comparación del retardo obtenido con distintas versiones del S.O. Linux Ubuntu, utilizando dispositivos Bluetooth 2.0 + EDR para un valor del parámetro ACL MTU de 512 octetos (Revisión de HCI: 0xc5c).	293
A.2. Comparación del retardo obtenido con distintas versiones del S.O. Linux Ubuntu, utilizando dispositivos Bluetooth 2.0 + EDR para un valor del parámetro ACL MTU de 1021 octetos (Revisión de HCI: 0xc5c).	294
A.3. Comparación del retardo obtenido, en distintas plataformas de medida, ambas con el S.O. Linux Ubuntu 9.10 Karmic Koala, utilizando dispositivos Bluetooth 2.0 + EDR para un valor del parámetro ACL MTU de 1021 octetos (Revisión de HCI: 0xc5c).	295

Índice de tablas

1.1. Productos Bluetooth oficialmente registrados en el Bluetooth SIG, agrupados por categorías (fuente: [SIG10]).	2
2.1. Evolución cronológica de la tecnología Bluetooth	8
2.2. Clases de dispositivos Bluetooth (adaptada de: [Bluetooth SIG04, parte A])	11
2.3. Tipos de paquetes ACL	19
2.4. Significado del campo LLID de la cabecera del <i>payload</i>	44
2.5. Tipos de tramas TS 07.10 utilizadas por RFCOMM	47
2.6. Perfiles Bluetooth (I)	48
2.7. Perfiles Bluetooth (II)	49
3.1. Taxonomía de los artículos sobre el rendimiento de Bluetooth	83
3.2. Herramientas de simulación empleadas en los distintos estudios	84
3.3. Taxonomía de los artículos que versan sobre la relación entre la tasa BER y la relación SNR, para distintas modulaciones	85
4.1. Longitudes máximas de los paquetes BB y nomenclatura asociada	97
4.2. Definición del parámetro m en función del tipo de modulación	97
4.3. Comparación de la tasa BER, la tasa PER y el retardo medio obtenidos en condiciones ideales y para una relación SNR de 10,5 dB, 12 dB y 15 dB, y 70 y 200 octetos de datos de usuario, en función del tipo de paquete empleado. . .	124
4.4. Tasas BER, para GFSK, correspondientes a las relaciones SNR empleadas en la figura 4.17.	127

5.1. Valores de los parámetros comunes a todas la medidas	135
5.2. Valores configurados del parámetro ACL MTU, tamaño de cada <i>buffer</i> del dispositivo Bluetooth, junto con el número de elementos de <i>buffer</i>	137
5.3. Valores del parámetro <i>Latency</i> (y del intervalo de <i>poll</i> al que equivale) para los que se ha evaluado el efecto de la configuración de la calidad de servicio	151
6.1. Resultados de las medidas de BER obtenidos para las modulaciones GFSK, $\pi/4$ DQPSK y 8 DQPSK, con una atenuación de señal de 45 dB.	244
6.2. Potencias de salida reales medidas para el rango de atenuación comprendido entre 0 dB y 10 dB y para un ancho de banda de medida (BW, <i>bandwidth</i>) de 2 MHz y de 80 MHz, en torno a la frecuencia central de 2441 MHz	247
6.3. Potencias de salida reales medidas para las frecuencias 2402 MHz, 2420 MHz, 2441 MHz, 2460 MHz y 2480 MHz	249
6.4. Error medio cometido en la estimación del retardo a partir de la tasa BER medida para valores de la relación SNR entre 10,95 y 14,95 dB, en ambos sentidos de transmisión.	264

Lista de acrónimos, abreviaturas y símbolos

Acrónimos y abreviaturas

ACK Acknowledge.

ACL (Bluetooth 1.1) Asynchronous Connection-Less.

ACL (Bluetooth 1.2) Asynchronous Connection-Oriented.

AFH Adaptive Frequency Hopping.

AM_ADDR Active Member Address.

AR_ADDR Access Request Address.

ARQ Automatic Repeat reQuest.

ARQN Automatic Repeat reQuest Number.

ASB Active Slave Broadcast.

AWGN Additive White Gaussian Noise.

B-Frame	Basic Information Frame.
BB	Baseband.
BCS	Bluetooth Carrier Sense.
BD_ADDR	Bluetooth Address.
BER	Bit Error Rate.
BNEP	Bluetooth Network Encapsulation Protocol.
BPSK	Binary Phase Shift Keying.
BW	Bandwidth.
CAC	Channel Access Code.
CRC	Cyclic Redundancy Check.
CSR	Cambridge Silicon Radio Plc.
DAC	Device Access Code.
DH	Data High-Rate.
DISC	Disconnect.
DM	Data Medium-Rate.
DM (RFCOMM)	Disconnected Mode.
DPSK	Differential PSK.
DQPSK	Differential Quaternary PSK.

DV	Data-Voice.
EDR	Enhanced Data Rate.
EIA	Electronic Industries Alliance.
eSCO	Extended Synchronous Connection-Oriented.
EV	Extended Voice.
FEC	Forward Error Correction.
GFSK	Gaussian Frequency Shift Keying.
HCI	Host Controller Interface.
HCTL	Host Controller Transport Layer.
HEC	Header Error Check.
HS	High Speed.
HV	High-quality Voice.
IAC	Inquiry Access Code.
ISI	Inter-Symbol Interference.
ISM	Industrial, Scientific and Medical.
ISS	Integrated System Solution Corp.
L2CAP	Logical Link Control and Adaptation Protocol.
L_CH	Logical Channel.

LE	Low Energy.
LLID	Logical Link Identifier.
LM	Link Manager.
LMP	Link Management Protocol.
LQ	Link Quality.
LSB	Least Significant Bit.
LT_ADDR	Logical Transport Address.
MSB	Most Significant Bit.
MTU	Maximun Transmission Unit.
N1	Maximum Frame Size.
NAK	Negative Acknowledge.
PAN	Personal Area Network.
PDU	Protocol Data Unit.
PER	Packet Error Rate.
PM_ADDR	Parked Member Address.
PRBS9	Pseudo-Random Bit Sequence.
PSB	Parked Slave Broadcast.
PSD	Power Spectral Density.

PSK	Phase Shift Keying.
QoS	Quality Of Service.
RF	Radio Frequency.
RFCOMM	Serial Cable Emulation Protocol basado en la norma ETSI TS 07.10.
RMS	Root Mean Square.
RSSI	Received Signal Strength Indicator.
SABM	Set Asynchronous Balanced Mode.
SAR	Segmentation And Reassembling.
SCO	Synchronous Connection-Oriented.
SD	Secure Digital.
SDP	Service Discovery Protocol.
SDU	Signalling Data Unit.
SEQN	Sequential Numbering scheme.
SIG	Special Interest Group.
SNR	Signal Noise Ratio.
SPI	Serial Peripheral Interface.
SPP	Serial Port Profile.
TC	Turbo Code.

TCP	Transport Control Protocol.
TCS	Telephony Control Protocol Specification.
TDD	Time Division Duplex.
TDM	Time Division Multiplexing.
TIA	Telecommunications Industry Association.
UA	Unnumbered Acknowledgement.
UART	Universal Asynchronous Receiver-Transmitter.
UIH	Unnumbered Information with Header check.
USB	Universal Serial Bus.
WLAN	Wireless Local Area Network.
WPAN	Wireless Personal Area Network.

Símbolos

δ	Offset que se incluye en el cálculo del retardo de transmisión para SPP en el sentido esclavo-maestro, y que afecta a longitudes de datos que exceden la capacidad de un paquete de 5 slots cuando tienen lugar retransmisiones.
$\Delta_{EDR}(m)$	Tiempo debido a la sobrecarga adicional añadida a los paquetes que utilizan las modulaciones EDR.
γ_b	Relación SNR por bit.
γ_s	Relación SNR por símbolo.
$\lceil Y \rceil$	Operador que redondea el valor de Y al menor entero superior a Y .
$\lfloor Y \rfloor$	Operador que redondea el valor de Y al mayor entero inferior a Y .
$\overline{N_{Tx}}$	Número medio de retransmisiones expresado en función de la tasa PER.
$\overline{t_{ACK_l}}(x)$	Promedio de $t_{ACK}(x)$ teniendo en cuenta el efecto de las retransmisiones.
$\overline{t_{R_{M_l}}}(N)$	Retardo medio para el perfil SPP en el sentido de transmisión maestro-esclavo, teniendo en cuenta las retransmisiones.
$\overline{t_{R_M}}(N)$	Retardo medio de transmisión para el perfil SPP, en el sentido maestro-esclavo.
$\overline{t_{R_{S_l}}}(N)$	Retardo para el perfil SPP en el sentido de transmisión esclavo-maestro, teniendo en cuenta las retransmisiones.
$\overline{t_{R_s}}(N)$	Retardo medio de transmisión para el perfil SPP, en el sentido esclavo-maestro.
$\overline{t_{TX_l}}(x)$	Promedio de $t_{TX}(x)$ teniendo en cuenta el efecto de las retransmisiones.

ρ	Coeficiente de correlación entre dos señales.
$\text{erfc}(z)$	Función error complementario.
$I_n(x)$	Función de Bessel modificada de orden n .
B	Ancho de banda de recepción, expresado en Hz.
C_{ff}^{USB}	Corrección introducida para modelar el efecto de la capa de transporte USB en la transferencia de una trama RFCOMM final.
C_{fnf}^{USB}	Corrección introducida para modelar el efecto de la capa de transporte USB en la transferencia de una trama completa de RFCOMM (1008 octetos).
$C_{TX}^{USB}(x)$	Corrección introducida para modelar el efecto de la transferencia a través de la capa de transporte USB en la transferencia de una trama RFCOMM única.
$d(m, x)$	Retardo de transmisión de x octetos de usuario, utilizando el modo EDR m .
E_b	Energía por bit transmitido, expresada en W.
E_s	Energía por símbolo transmitido, expresada en W.
h	Índice de modulación para FSK.
$H_{df}(x)$	Tamaño de la cabecera de la cabecera del campo de datos (<i>Data Field</i>).
$H_{df}^{DH}(x)$	Tamaño de la cabecera de la cabecera del campo de datos (<i>Data Field</i>) para paquetes DH.
$H_{df}^{DM}(x)$	Tamaño de la cabecera de la cabecera del campo de datos (<i>Data Field</i>) para paquetes DM.
H_L	Tamaño de la cabecera de una trama del modo básico del protocolo L2CAP en Bluetooth v1.2 y posteriores, equivalente a la PDU de L2CAP en Bluetooth 1.1 (4 octetos).

L_1	Máximo número de octetos que pueden transportarse en el campo de datos de un paquete Bluetooth de 1 <i>slot</i> .
L_3	Máximo número de octetos que pueden transportarse en el campo de datos de un paquete Bluetooth de 3 <i>slots</i> .
L_5	Máximo número de octetos que pueden transportarse en el campo de datos de un paquete Bluetooth de 5 <i>slots</i> .
$L_{ff}(x)$	Longitud del fragmento final RFCOMM resultante de dividir los x octetos de datos de usuario en fragmentos de L_R octetos.
$L_{m,s}$	Número de octetos de un paquete DH que ocupa s <i>slots</i> transmitido a m Mbps.
L_R	Longitud a la que RFCOMM ha de fragmentar los datos de usuario, garantizando, por un lado, que no se excede la MTU de L2CAP (M_R), y por otro, que se respeta el valor máximo fijado por el parámetro $N1$, teniendo en cuenta la máxima sobrecarga introducida por el protocolo RFCOMM.
M	Número de mensajes o símbolos de la constelación de la modulación considerada.
$M(x)$	Número de octetos de una trama de L2CAP, incluyendo la sobrecarga añadida por los niveles RFCOMM y L2CAP.
M_{ACL}	Parámetro ACL MTU, expresado en octetos, que representa el tamaño de los <i>buffers</i> del dispositivo Bluetooth.
M_{BB}	Mínimo entre el parámetro ACL MTU, M_{ACL} y la máxima capacidad de un paquete del nivel BB de 5 <i>slots</i> , L_5 .
M_{USB}	Tamaño al que se produce la fragmentación interna para la transferencia a través de la capa de transporte HCTL USB: 512 octetos.
M_R	MTU de L2CAP para RFCOMM.

N_0	Densidad espectral del ruido en un canal AWGN, expresada en W/Hz.
$n_{\text{rff}}(x)$	Número de fragmentos RFCOMM no finales necesarios para transmitir x octetos de datos de usuario.
n_{poll}	Intervalo de tiempo, en <i>slots</i> , entre <i>poll</i> consecutivos.
$n_{\text{sym}}(m, x)$	Número total de símbolos transmitidos para x octetos, utilizando la modulación indicada por el parámetro m .
$n_b(x)$	Número total de bits transmitidos para x octetos, en Bluetooth 1.1.
$n_d(x)$	Número de bits de los que consta el campo de datos (<i>Data Field</i>), incluyendo la cabecera, los correspondientes a los x octetos del cuerpo del <i>payload</i> y a los 2 octetos de CRC y, en el caso de los paquetes DM, los bits de redundancia añadidos por la codificación FEC 2/3.
$n_g(x)$	Número de grupos de 15 bits de los que consta un paquete DM tras haber aplicado la codificación FEC 2/3 al número total de bits computados a partir de $n_d(x)$ para x octetos de datos.
O_{BB}	Tamaño de la sobrecarga del paquete Bluetooth, definido como la suma de los bits de la cabecera del paquete (54 bits) y del código de acceso (72 bits), y que supone un total de 126 bits.
O_{CRC}	Sobrecarga asociada al campo CRC de los paquetes Bluetooth.
$O_{\text{R}}(x)$	Tamaño de la sobrecarga introducida por el protocolo RFCOMM (4 ó 5 octetos) para x octetos de datos de usuario.
P_b	Probabilidad de error de bit.
$P_{\text{DH}_d}(x)$	Probabilidad de que el campo de datos de un paquete DH, que transporta x octetos, no contenga ningún bit erróneo, expresada en función de la tasa BER.

$P_{DM_d}(x)$	Probabilidad de que el campo de datos de un paquete DM, que transporta x octetos, no contenga errores irreversibles, expresada en función de la tasa BER.
P_g	Probabilidad de que un grupo de 15 bits de un paquete DM no contenga errores irreversibles.
P_H	Probabilidad de que no se produzca un error irreversiblemente en la cabecera de un paquete DH o DM.
$P_l(x)$	Tasa PER para paquetes DH y DM expresada en función de la tasa BER y teniendo en cuenta que los errores se pueden producir en la cabecera del paquete o en el campo de datos, en el que se transportan x octetos.
$P_{nl}(x)$	Probabilidad de que no se pierda un paquete DH o DM, es decir, de que no se produzca un error irreversiblemente ni en la cabecera ni en el campo de datos en el que se transportan x octetos.
P_r	Potencia de señal recibida a la entrada del receptor, expresada en W.
P_s	Probabilidad de error de símbolo.
$Q(z)$	Probabilidad de que una variable gaussiana con media 0 y varianza 1 sea mayor que z .
$Q_1(a, b)$	Función Q de Marcum.
$t_{ACK}(x)$	Tiempo empleado en transmitir y confirmar cada uno de los paquetes del nivel BB necesarios para enviar x octetos de datos.
$t_{ACK}^{ACL}(x)$	Modificación de la componente $t_{ACK}(x)$ para incluir el efecto de la fragmentación según el valor del parámetro ACL MTU.
$t_{ACK}^c(x)$	Modificación de la componente $t_{ACK}(x)$ para incluir el efecto de la capa de transporte y de la fragmentación según el valor del parámetro ACL MTU.

t_{gEDR}	Tiempo de guarda EDR (valor comprendido entre 4,75 μs y 5,25 μs).
T_{poll}	Intervalo de tiempo, en μs , entre <i>poll</i> consecutivos.
$t_{RTx_{max}}(x)$	Cota máxima de $t_{RTx}(x)$.
$t_{RTx_{min}}(x)$	Cota mínima de $t_{RTx}(x)$.
$t_{RTx}(x)$	Tiempo entre retransmisiones consecutivas de un paquete Bluetooth que transporta x octetos en el campo de datos.
t_{sEDR}	Tiempo de la secuencia de sincronismo EDR (11 μs).
t_{sym}	Tiempo de transmisión de un símbolo (1 μs).
t_{tEDR}	Tiempo de cola EDR (2 μs).
$t_{TX}(x)$	Tiempo empleado en la transmisión de x octetos de datos sin considerar el tiempo de confirmación del último (o único) paquete de nivel BB en el que se transporta el último (o único) fragmento.
$t_{TX}^{ACL}(x)$	Modificación de la componente $t_{TX}(x)$ para incluir el efecto de la fragmentación según el valor del parámetro ACL MTU.
$t_{TX}^c(x)$	Modificación de la componente $t_{TX}(x)$ para incluir el efecto de la capa de transporte y de la fragmentación según el valor del parámetro ACL MTU.
$t_{USB}(x)$	Retardo introducido por la capa de transporte HCTL USB en dispositivos del fabricante CSR.
t_b	Tiempo de transmisión de un bit: 1 μs , en Bluetooth 1.1.
T_b	Periodo de transmisión de un bit, expresado en segundos.
T_s	Duración de un <i>slot</i> : 625 μs .
T_s	Periodo de transmisión de un símbolo, expresado en segundos.
$x \text{ mód } y$	Operador que obtiene el resto de la división entera de los operandos x e y .

CAPÍTULO 1

Introducción

La tecnología Bluetooth, hoy por hoy, continúa siendo la tecnología inalámbrica más extendida para la formación de redes de área personal (PAN, *Personal Area Networks*), siendo cada vez más las aplicaciones para las que se considera su utilización, incluso desde la fase de especificación, a través de la definición de perfiles. Precisamente, el hecho de que el estándar incluya la definición de los perfiles, en los que se establecen los requisitos que un dispositivo debe satisfacer para ofrecer un determinado servicio, así como los protocolos a utilizar y los procedimientos a seguir en distintos escenarios de aplicación, permite garantizar la interoperatividad entre dispositivos de distintos fabricantes y, por ende, su usabilidad, lo que propicia la gran difusión comercial que ostenta esta tecnología.

Así, desde la aparición en el mercado de los primeros dispositivos Bluetooth en el año 2000 (teléfonos móviles, auriculares o *headsets*, adaptadores para PC, etc.) ha ido aumentando la diversidad de dispositivos dotados con tecnología Bluetooth, llegándose a integrar no sólo en periféricos de ordenadores personales (teclados, ratones e impresoras), sino también en receptores GPS, cámaras digitales, reproductores MP3, relojes, consolas de juego como la *Wii* o la *PlayStation 3*, e incluso en sensores médicos, tal y como puede comprobarse en la tabla 1.1. Concretamente, en dicha tabla se muestran las categorías de productos y número de unidades que han superado el proceso de certificación necesario para poder comercializarse utilizando la marca Bluetooth. No obstante, cabe señalar que,

Tabla 1.1.: Productos Bluetooth oficialmente registrados en el Bluetooth SIG, agrupados por categorías (fuente: [SIG10]).

Categoría	Unidades	Categoría	Unidades
Audio/Visual	2259	Dispositivos médicos	124
Automoción	2029	Teléfonos	3363
Dispositivos de juego	84	Accesorios de telefonía	1850
Ordenadores de mano	1092	Equipos de oficina	762
Auriculares	1370	Ordenadores personales	1557
Dispositivos domésticos	661	Productos únicos	113
Dispositivos de entrada	329		
Número total de productos		11152	

además de estos productos registrados por el Bluetooth SIG, se encuentran disponibles en el mercado otros dispositivos que, aunque son compatibles con la tecnología Bluetooth, no han sido certificados.

Adicionalmente, a la alta difusión comercial lograda se suma el hecho de que la especificación del sistema Bluetooth, al tiempo que es objeto de investigación incluso doce años después de sus orígenes, continúa actualizándose y ampliando para diversificarse y abarcar aplicaciones para las que esta tecnología inicialmente no fue concebida. Es el caso de la última versión, Bluetooth 4.0 [Bluetooth SIG10], publicada recientemente el 30 de junio de 2010, que abarca tanto la especificación de dispositivos Bluetooth de alta velocidad (HS, *High Speed*) como de dispositivos de consumo ultrareducido (LE, *Low Energy*).

De todos los perfiles contemplados en el estándar, en esta tesis se ha optado por estudiar el perfil SPP (*Serial Port Profile*), para Bluetooth 2.0 + EDR, debido a que la mayoría de los dispositivos Bluetooth, desde terminales móviles y receptores GPS hasta sensores médicos comerciales, de fabricantes distintos (Nonin Medical [Nonin Medical], CorScience [CorScience] y A&D Medical [A&D Medical]), implementan el perfil SPP. Asimismo, el protocolo en el que se basa este perfil, RFCOMM (*Serial Cable Emulation Protocol*, basado en la norma ETSI TS 07.10 [ETSI97]), es también la base de otros perfiles entre los que cabe citar como ejemplos: el perfil de acceso remoto a redes (DUN, *Dial-up Networking Profile*),

el perfil de FAX, el perfil de intercambio de objetos (OBEX, *OBject EXchange*), el perfil de transferencia de ficheros (FTP, *File Transfer Profile*), el perfil de auricular (HSP, *Headset Profile*) y el perfil de control remoto de audio/video (AVRCP, *Audio/Video Remote Control Profile*).

Por tanto, el objetivo principal de la presente tesis ha sido modelar el retardo de transmisión para el perfil de puerto serie SPP en Bluetooth 2.0 + EDR, tanto para el modo básico, compatible con Bluetooth 1.1, como para el modo EDR (*Enhanced Data Rate*).

El estudio desarrollado ha consistido en formular expresiones matemáticas para calcular el retardo de transmisión, en los sentidos maestro-esclavo y esclavo-maestro, teniendo en cuenta la sobrecarga agregada por los protocolos que intervienen en la transmisión de la información de usuario al emplear el perfil SPP.

La estimación teórica del retardo de transmisión se ha llevado a cabo, en una fase inicial, asumiendo condiciones ideales. Sin embargo, el hecho de que Bluetooth opere en la banda ISM, compartida con otras tecnologías inalámbricas, convierte a los escenarios sin pérdidas en poco realistas. Por este motivo, se ha considerado imprescindible completar el modelo sin pérdidas para contemplar el efecto de las retransmisiones en el cálculo del retardo asociado al perfil SPP. Para ello se ha determinado el retardo de transmisión a partir de la tasa de error de bit (BER, *Bit Error Rate*), para cada una de las modulaciones, en función de la relación señal/ruido, SNR (*Signal Noise Ratio*), asumiendo para ello que el canal de transmisión Bluetooth se puede modelar como un canal de ruido blanco gaussiano (AWGN, *Additive White Gaussian Noise*), es decir, considerando que el efecto del ruido es equivalente a la adición de ruido gaussiano de potencia constante en la banda considerada.

Posteriormente, para validar empíricamente el modelo desarrollado se ha utilizado la pila de protocolos BlueZ. En el caso de la evaluación del modelo sin considerar la ocurrencia de retransmisiones, se han empleado dispositivos reales de distintos fabricantes de gran implantación en productos Bluetooth comerciales: CSR (*Cambridge Silicon Radio*), ISS (*Integrated System Solution Corp.*) y *Broadcom Corporation*. Al respecto cabe resaltar que, precisamente, CSR, desde el lanzamiento de su primer *chipset* Bluetooth *BlueCore* [CSR00] en el año 2000, junto con el *kit* de desarrollo *Casira* [CSR10], ha llegado a suministrar en torno al 70% de los *chipset* Bluetooth en el año 2005 [Finch05]. De hecho, en [Finch05] se explica que no sólo ha llegado a liderar el mercado de consumo, sino que también se

ha expandido en el mercado industrial. No obstante, desde el año 2007, aunque sigue conservando esta posición de liderazgo, manteniendo una cuota de mercado en torno al 50 % [Reuters07], como principal competidor comienza a destacar el fabricante *Broadcom*, de forma que entre ambos suman el 75 % del mercado [Electronics07].

En lo que respecta a la validación del modelo considerando el efecto de las retransmisiones, llevada a cabo exclusivamente con dispositivos del fabricante CSR integrados en placas de desarrollo, del fabricante BlueGiga, se ha abordado en las siguientes fases:

- Se ha diseñado un escenario que permite reproducir empíricamente el modelo de canal gaussiano propuesto teóricamente, en el que se considera que las transmisiones entre dos dispositivos Bluetooth sólo se ven afectadas por un factor de atenuación y la adición de un nivel de ruido blanco gaussiano.
- Se ha obtenido la tasa BER, comparándola con la prevista teóricamente a partir de la relación SNR.
- Se ha concluido con la comparación del retardo medido con el retardo calculado analíticamente a partir de la tasa BER medida y computada en distintos casos.

La presente memoria, en la que se expone el estudio realizado, consta de siete capítulos (incluyendo el presente capítulo introductorio), en cada uno de los cuáles se abordan los siguientes contenidos:

En el capítulo 2 se presenta una breve introducción a la tecnología Bluetooth, comenzando por Bluetooth 1.1 y sintetizando los cambios introducidos en versiones posteriores, hasta Bluetooth 2.1 + EDR.

En el capítulo que versa sobre el Estado de la Técnica, capítulo 3, se realiza una breve y sumaria exposición de los trabajos de investigación que se centran en estudios de rendimiento de Bluetooth.

En el capítulo 4 se expone el modelo desarrollado, empezando por formular el cálculo del retardo de transmisión para el perfil SPP en Bluetooth 1.1, desarrollado originalmente en

[Moron08]. Después se aborda la extensión del modelo para Bluetooth 2.0 + EDR, sin tener en cuenta la ocurrencia de retransmisiones (sección 4.3) y considerándolas (sección 4.4).

En el capítulo 5 se exponen las medidas realizadas con dispositivos de distintos fabricantes para evaluar empíricamente el modelo teórico desarrollado suponiendo condiciones ideales, es decir, que no se producen pérdidas de paquetes que den lugar a retransmisiones. Adicionalmente, el capítulo 5 se complementa con el Anexo A, en el que se incluyen algunos de los resultados obtenidos bajo las últimas versiones de Linux, comparándolos con los de las versiones con las que se ha llevado a cabo la validación empírica.

Utilizando un escenario con pérdidas, en el capítulo 6 se expone la validación empírica del modelo de retardo teórico para el perfil SPP teniendo en cuenta el efecto de las retransmisiones.

Finalmente, en el capítulo 7, se resumen las conclusiones extraídas y se plantean las líneas futuras de investigación.

CAPÍTULO 2

Fundamentos teóricos de Bluetooth

Bluetooth es una tecnología radio de corto alcance, concebida para reemplazar los cables de conexión entre dispositivos electrónicos fijos o móviles, y que permite la transferencia de voz y datos. Las características que la definen son: robustez, baja complejidad, así como consumo y coste reducidos.

Bluetooth opera en la banda ISM (*Industrial, Scientific and Medical*) a 2,4 GHz y, para aumentar la robustez de la conexiones o, lo que es lo mismo, minimizar el impacto de interferencias y desvanecimientos, utiliza una técnica de espectro ensanchado por salto de frecuencia (*frequency hopping*). Esta técnica consiste en sintonizar el transceptor radio, del transmisor y del receptor, a una frecuencia seleccionada de forma pseudoaleatoria, en cada transmisión.

Desde que el grupo Bluetooth SIG (*Special Interest Group*), fundado en el año 1998 por cinco compañías, desarrollara la versión 1.0b de la especificación de la tecnología Bluetooth, publicada en el año 1999, durante doce años se ha continuado evolucionando dicha especificación, pasando por todas las fases indicadas en la tabla 2.1, hasta alcanzar las últimas versiones, Bluetooth 3.0 + HS y Bluetooth 4.0, publicadas en los meses de abril de 2009 y junio de 2010, respectivamente.

Tabla 2.1.: Evolución cronológica de la tecnología Bluetooth

Año	Evento
1999	Publicación de la versión Bluetooth 1.0b
2001	Publicación de la versión Bluetooth 1.1 [Bluetooth SIG01]
2002	Bluetooth 1.1 es parcialmente adoptada por el IEEE bajo el estándar IEEE 802.15.1-2002 [IEEE 802.15 02]
2003	Publicación de la versión Bluetooth 1.2 [Bluetooth SIG03]
2004	Publicación de la versión Bluetooth 2.0 + EDR [Bluetooth SIG04]
2005	Revisión del estándar IEEE 802.15.1, basada en Bluetooth 1.2: IEEE 802.15.1-2005 [IEEE 802.15 05]
2007	Publicación de la versión Bluetooth 2.1 + EDR [Bluetooth SIG07]
2009	Publicación, en el mes de abril, de la versión Bluetooth 3.0 + HS [Bluetooth SIG09]
2010	Publicación, en el mes de junio, de la versión Bluetooth 4.0 [Bluetooth SIG10]

Aunque el modelo desarrollado en la presente tesis se ha llevado a cabo con dispositivos Bluetooth 2.0 + EDR y Bluetooth 2.1 + EDR, en el apartado 2.1 de este capítulo, se comienza presentado la tecnología Bluetooth, partiendo de la especificación 1.1, para continuar exponiendo los cambios introducidos en las versiones posteriores 1.2, 2.0 y 2.1, que se exponen en el apartado 2.2.

Finalmente, en la sección 2.3 se realiza una breve introducción de los perfiles definidos en el estándar Bluetooth, dedicando especial atención al perfil SPP en el que se centra el estudio abordado.

2.1. Bluetooth 1.1

El sistema Bluetooth permite tanto el establecimiento de conexiones punto a punto entre dos dispositivos Bluetooth, como de conexiones punto-multipunto.

En la conexión punto-multipunto el canal es compartido por varias unidades Bluetooth, formándose lo que se denomina una *piconet*. En una *piconet* sólo una unidad desempeña

el rol de maestro, mientras que las restantes, hasta un máximo de siete, participan como esclavos. Este límite se refiere al número de esclavos que pueden encontrarse en modo activo, ya que la especificación contempla que en otro modo de funcionamiento —PARK (véase el apartado 2.1.2.3)—, aunque no activos, si puedan estar sincronizados al canal de la piconet un número prácticamente ilimitado. En cualquier caso, independientemente del modo de operación en el que se encuentre un esclavo, el acceso al canal siempre es controlado por el maestro.

El canal de una *piconet* soporta transmisiones *full-duplex*, para lo que se usa un esquema TDD (*Time Division Duplex*).

Bluetooth contempla una segunda topología de red, *scatternet*, a la que da lugar el solapamiento de varias *piconets*. Esto es, una *piconet* puede tener un único maestro, pero un esclavo puede participar en distintas *piconets*, pudiendo adoptar en una segunda *piconet* el papel de maestro o de esclavo. Al respecto, nótese que un esclavo puede pertenecer a varias *piconets* con el mismo rol, pero sólo en una podrá actuar como maestro. En una *scatternet*, las diferentes *piconets* que la integran no están sincronizadas en frecuencia, es decir, cada *piconet* tiene su propia secuencia de salto, determinada a partir de parámetros del maestro, según se explica en el apartado 2.1.2.1.

El sistema Bluetooth, representado en la figura 2.1, consta de:

- Transceptor Radio o nivel *Radio Frequency* (RF), cuyas características se indican en el apartado 2.1.2.
- Nivel *Baseband* (BB), descrito en el apartado 2.1.2.
- Nivel *Link Manager* (LM), presentado en el apartado 2.1.3.
- Arquitectura de protocolos propiamente dicha, en la que la capa inferior es el nivel L2CAP (*Logical Link Control and Adaptation Protocol*), introducido en el apartado 2.1.5.

Los tres primeros componentes —RF, BB y LM— integran el denominado subsistema *Bluetooth Host Controller* o Controlador Bluetooth, referido normalmente como *Bluetooth Hardware* o módulo Bluetooth. Asimismo, la arquitectura de protocolos se encapsula en el subsistema *Host* o *Bluetooth Host*. Adicionalmente, para que el *Host* pueda acceder a las

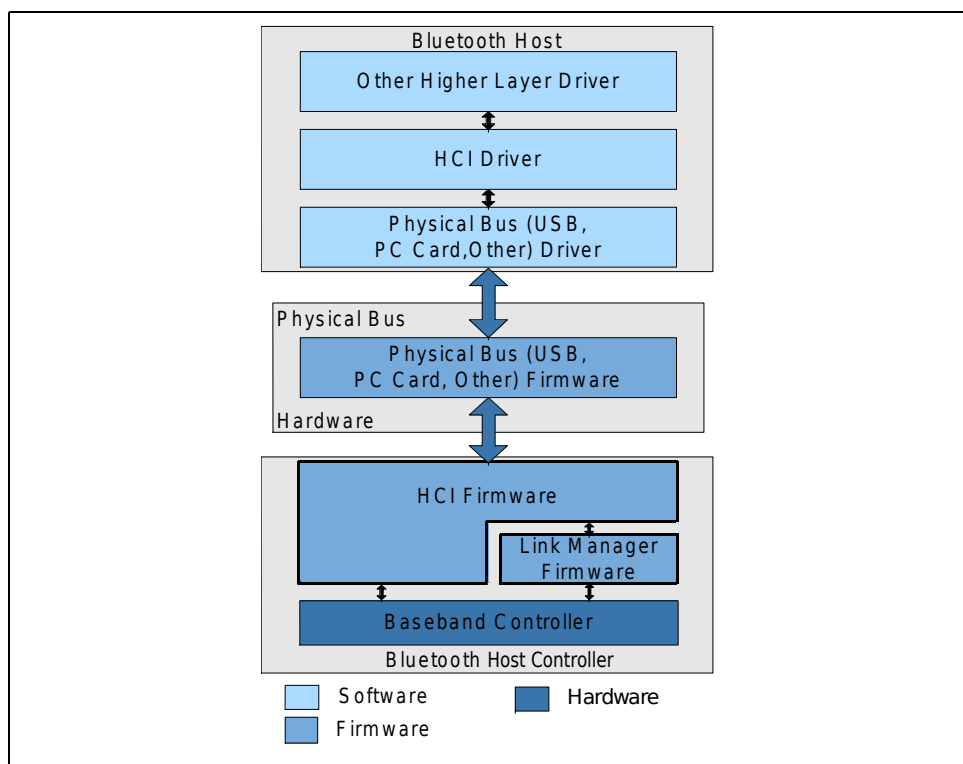


Figura 2.1.: Arquitectura de protocolos Bluetooth 1.1. (adaptada de: [Bluetooth SIG01, parte H:1])

capacidades del subsistema *hardware*, se define una interfaz física estándar denominada HCI (*Host Controller Interface*), que está integrada por tres componentes: *HCI Driver*, *HCTL* (*Host Controller Transport Layer*) y *HCI Firmware*.

La definición del nivel HCI, presentado en el apartado 2.1.4, de uso opcional, garantiza la interoperatividad entre los subsistemas *Bluetooth Controller* y *Host* cuando no están integrados en un mismo dispositivo.

2.1.1. Nivel RF (*Radio Frequency*)

El subsistema radio Bluetooth tiene las siguientes características:

- Utiliza la banda ISM a 2,4GHz, definiendo un total de 79 (ó 23 canales, en el caso de países que no dispongan de toda la banda) de 1 MHz de ancho de banda.

- La transmisión se realiza a 1 Mbit/s, empleando la modulación GFSK (*Gaussian Frequency Shift Keying*), que simplifica el diseño del transceptor radio.
- Para mejorar la robustez de la transmisión, utiliza la técnica de espectro ensanchado por salto de frecuencia (*frequency hopping*), realizando 1600 saltos de canal por segundo (una vez cada 625 μ s), siguiendo un esquema pseudoaleatorio.
- La potencia máxima transmitida está determinada por la clase del dispositivo, según se muestra en la tabla 2.2. Para mejorar el rango de uso, los dispositivos, opcionalmente, pueden incorporar control de potencia de salida.

Tabla 2.2.: Clases de dispositivos Bluetooth (adaptada de: [Bluetooth SIG04, parte A])

Clase	Potencia máxima	Potencia nominal	Potencia mínima	Control de potencia
1	100 mW (20 dBm)	—	1 mW (0 dBm)	Obligatorio
2	2,5 mW (4 dBm)	1 mW (0 dBm)	0,25 mW (-6 dBm)	Opcional
3	1 mW (0 dBm)	—	—	Opcional

2.1.2. Nivel BB (*Baseband*)

Las funciones principales del nivel BB son las siguientes:

- Control de la temporización de transmisión y recepción.
- Determinación de la secuencia de salto que define el canal de la *piconet*.
- Generación de las claves necesarias para los procedimientos de autenticación y cifrado.
- Composición y descomposición en paquetes Bluetooth de los datos de las capas superiores (L2CAP y LM).

- Corrección y detección de errores.
- Control de flujo de la transmisión.

Con el objetivo de dar una visión global del funcionamiento del nivel BB, en los apartados 2.1.2.1 y 2.1.2.2 se comienza definiendo los conceptos de canal físico y de enlace físico. A continuación, en el apartado 2.1.2.3 se resume la máquina de estados que rige el funcionamiento del nivel BB y se presentan los distintos modos de operación contemplados en la especificación. Por último, en el apartado 2.1.2.4 se describe la estructura del paquete Bluetooth y se presentan los tipos de paquetes definidos para cada uno de los enlaces físicos y, finalmente, en el apartado 2.1.2.5 se describen los mecanismos de corrección de errores proporcionados por el nivel BB.

2.1.2.1. Canal físico

El canal físico está representado por una secuencia de salto pseudoaleatoria de 79 ó 23 canales RF, única para cada *piconet*, ya que se obtiene a partir de la dirección Bluetooth y del reloj del maestro. Esta información, comunicada a los esclavos durante el procedimiento de conexión, impone la fase de dicha secuencia que define el canal una vez establecida la conexión. A su vez, el canal está dividido en ranuras temporales de $625 \mu s$ de duración, denominadas *time slots*, de forma que a cada *slot* corresponde una frecuencia RF diferente, siendo la frecuencia nominal de salto de 1600 saltos/s.

La compartición del canal de la *piconet* por parte del maestro y de los esclavos se basa en un esquema TDD por lo que todas las unidades de una misma *piconet* han de estar sincronizadas en tiempo y en frecuencia. De acuerdo con este esquema, el maestro y los esclavos alternan sus transmisiones, teniendo en cuenta que los *slots* se numeran cíclicamente de acuerdo con el reloj del maestro de la *piconet* comenzando desde 0 hasta $2^{27} - 1$, de modo que un maestro comienza su transmisión en *slots* pares y, un esclavo en *slots* impares.

Finalmente, en relación con la secuencia de salto, es conveniente matizar que para paquetes que ocupan más de un *slot*, la frecuencia RF de trabajo se mantiene durante toda la extensión del paquete, según se ilustra en la figura 2.2, donde puede observarse que, por ejemplo, si un paquete de tres *slots* comienza a transmitirse a la frecuencia $f(k)$, la siguiente frecuencia a emplear, en el *slot* de recepción, será $f(k + 3)$ y no $f(k + 1)$.

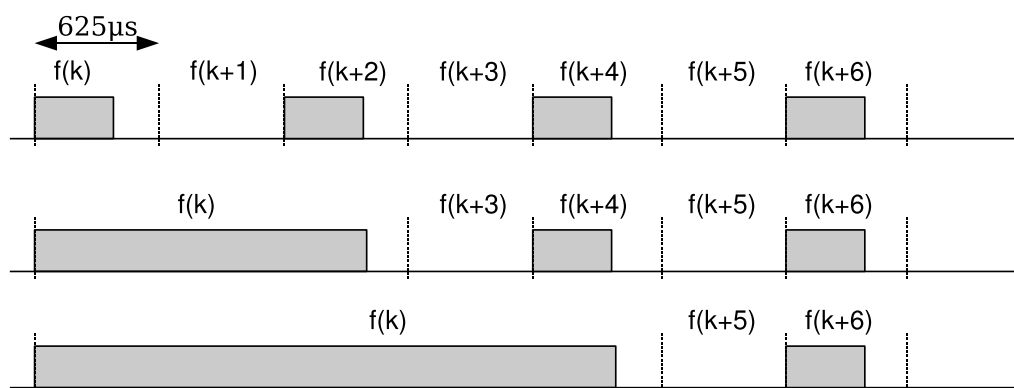


Figura 2.2.: Esquema de transmisión de paquetes *multi-slot* (fuente: [Bluetooth SIG01, parte B])

2.1.2.2. Enlaces físicos

Bluetooth define a nivel BB dos tipos de enlaces físicos entre dispositivos:

- **SCO** (*Synchronous Connection-Oriented*): Se define como un enlace simétrico punto a punto entre el maestro y un esclavo de una *piconet*. Tiene como objeto transmitir información síncrona, normalmente audio, para lo cual se realiza una reserva de *slots* de transmisión. Por consiguiente, se puede considerar una conexión de conmutación de circuitos entre ambos dispositivos.

Una unidad que actúa como maestro puede soportar hasta tres enlaces SCO con un sólo esclavo o con varios. Por contra, un esclavo puede tener establecidos también tres conexiones SCO con un único maestro y un máximo de dos en diferentes *piconets*.

- **ACL** (*Asynchronous Connection-Less*): En este caso se define como enlace punto a multipunto entre el maestro y todos los esclavos de la *piconet*. Se utiliza para el envío de información asíncrona que puede ocupar los *slots* disponibles, no utilizados para conexiones SCO. Se puede entender como una conexión de conmutación de paquetes entre todos los dispositivos de la *piconet*.

En relación con un enlace ACL existen las siguientes consideraciones:

- Entre un maestro y un esclavo sólo puede existir un único enlace ACL.

- Un esclavo sólo puede transmitir un paquete ACL si ha sido direccionado por el maestro en el *slot* anterior.
- Siempre que no haya datos pendientes de envío para un determinado enlace ACL y que no se haya cumplido el denominado intervalo de *poll* (véase el apartado 2.1.3), no tendrá lugar transmisión alguna.

2.1.2.3. Estados y modos de operación

El diagrama de estado de la figura 2.3 ilustra los diferentes estados en los que puede encontrarse el nivel BB. Existen dos estados principales: *STANDBY* y *CONNECTION*; además, hay siete subestados —*inquiry*, *inquiry scan*, *inquiry response*, *page*, *page scan*, *slave response* y *master response*— que son estados temporales por los que se pasa para constituir o extender una *piconet*. La transición entre los distintos estados se puede producir en respuesta a peticiones del LM, porque se complete un procedimiento satisfactoriamente, o por la expiración de algún temporizador.

El estado *STANDBY* es un estado de bajo consumo en el que se encuentra, por defecto, una unidad Bluetooth. A partir de este estado puede transitarse a cada uno de los siguientes subestados en las condiciones que se indican en los siguientes epígrafes, en los que también se describe cómo estos subestados permiten evolucionar desde el estado *STANDBY* al estado *CONNECTION*:

- ***inquiry***: Se transita a este subestado cuando una unidad Bluetooth quiere buscar otros dispositivos existentes en su zona de cobertura, con el fin de obtener la dirección Bluetooth, *BD_ADDR* (*Bluetooth Address*) y la información de reloj de un dispositivo concreto o de los dispositivos de una determinada clase para poder continuar estableciendo una conexión.

Para ello, el nivel BB permanece transmitiendo paquetes ID que contienen un código genérico o específico de una clase de dispositivos (véase el apartado 2.1.2.4), mientras que no se agote el tiempo de espera de respuestas o hasta que el nivel LM solicite la interrupción del procedimiento.

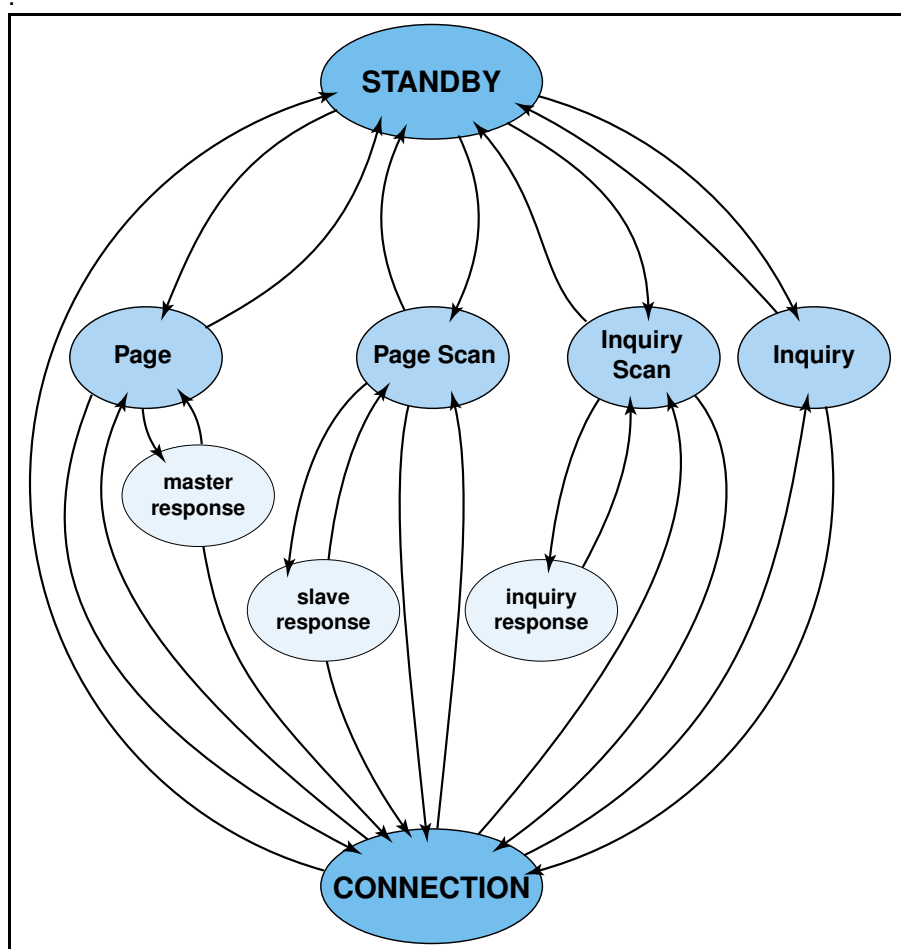


Figura 2.3.: Diagrama de estados definido en Bluetooth 1.1 (fuente: [Bluetooth SIG01, parte B])

- ***inquiry scan***: Un dispositivo permanece a la escucha de recibir mensajes procedentes de una unidad que está realizando un procedimiento de *inquiry*.

Al recibir un mensaje de *inquiry*, el nivel BB pasa al subestado *inquiry response* para responder.

- ***page***: Se transita a él en caso de que una unidad desee iniciar el procedimiento de establecimiento de conexión para incorporar un esclavo a la *piconet* de la que es maestro.

En este subestado el nivel BB transmite un mensaje de PAGE, es decir, un paquete ID con un código específico del dispositivo con el que desea conectarse, hasta que dicho dispositivo conteste o hasta que se exceda la temporización correspondiente.

En caso de que el esclavo responda, se transita al subestado *master response* en el que el maestro envía al esclavo un paquete de control —paquete FHS (véase el apartado 2.1.2.4)— con todos los parámetros necesarios para que éste pueda sincronizarse con la *piconet* y con la dirección AM_ADDR (*Active Member Address*) que se le asigna, utilizada para identificar los paquetes intercambiados con ese esclavo.

El maestro tiene que continuar retransmitiendo dicho paquete hasta recibir respuesta del esclavo o hasta que se agote el tiempo máximo de espera. En este último caso, se vuelve al subestado *page*, pero si el extremo remoto confirma su recepción, se culmina el procedimiento de establecimiento de conexión enviando un paquete POLL (véase el apartado 2.1.2.4) por el canal de la *piconet* que, para transitar al estado *CONNECTION*, debe ser confirmado antes de que transcurran un número determinado de *slots*. De no ser así, el nivel BB retorna al subestado *page*.

- **page scan:** Estado en el que se encuentra un dispositivo cuando permanece a la espera de que se produzca una conexión entrante, es decir, de recibir un mensaje de PAGE. Tras recibir dicho mensaje y responder, transita al subestado *slave response*. En este nuevo subestado, el esclavo aguarda la llegada del paquete FHS procedente del maestro mientras no expire el temporizador correspondiente, de forma que si se agota el tiempo de espera se retorna al subestado *page scan*.

En caso contrario, es decir, si recibe el citado paquete de control, después de confirmarlo, la transición al estado *CONNECTION* queda supeditada a la recepción y confirmación del paquete POLL enviado por el maestro.

En el estado *CONNECTION* el nivel BB de la unidad Bluetooth puede operar en uno de los siguientes modos:

- **ACTIVE:** En este modo de operación el dispositivo Bluetooth participa activamente en la *piconet*. El maestro planifica la transmisión en función del tráfico intercambiado con cada uno de los esclavos, teniendo en cuenta que debe realizar transmisiones

periódicas a cada uno de los esclavos para mantenerlos sincronizados a la *piconet*, independientemente de que haya datos pendientes de transmisión o no. Por su parte, los esclavos sólo pueden transmitir cuando han sido direccionados en el *slot* anterior, es decir, cuando han recibido un paquete con la dirección AM_ADDR que les ha sido asignada en el proceso de conexión. En caso de que un esclavo no haya sido direccionado, no está obligado a permanecer escuchando durante el resto de los *slots* que ocupe el paquete dirigido a otra unidad.

- **SNIFF:** En este modo de bajo consumo el esclavo reduce su actividad de escucha. Para ello, el maestro sólo puede iniciar su transmisión en ciertos *slots* —*sniff slots*— espaciados un periodo de tiempo (T_{sniff}) previamente acordado por los niveles LM del esclavo y el maestro. En dichos *slots*, el esclavo inicia la escucha durante un número de *slots* (N_{sniff}), también negociado a nivel de LM, de forma que si es direccionado, continua escuchando siempre que no se haya superado el número de N_{sniff} *slots*, o no expire el temporizador correspondiente. De hecho, mientras que no se exceda este temporizador el esclavo permanece escuchando siempre que continúe siendo direccionado.
- **HOLD:** Este modo de funcionamiento afecta únicamente a los enlaces ACL, de forma que se continúa soportando los enlaces SCO establecidos. El objetivo de este modo es permitir participar en otra *piconet* o realizar procedimientos de *inquiry*, *inquiry scan*, *page* o *page scan*.

Antes de entrar en el modo HOLD, maestro y esclavo negocian el tiempo —*holdTO*— en la que la conexión va a permanecer en este modo. Transcurrido este tiempo, el esclavo se despierta y se sincroniza al canal.

- **PARK:** Es un modo de bajo consumo al que pasa una conexión con un esclavo que no necesita participar activamente en la *piconet*, pero que sí desea continuar sincronizado al canal. En este modo, el esclavo pierde la dirección AM_ADDR previamente asignada, que es reemplazada por dos identificadores recibidos en el proceso de negociación que tiene lugar a nivel de LM:
 - PM_ADDR (*Parked Member Address*): Identificador que el maestro utilizará para direccionar al esclavo cuando desee iniciar el proceso de salida de este modo.

- AR_ADDR (*Access Request Address*): Dirección que permitirá al maestro identificar al esclavo que está cursando una petición de acceso al canal.

2.1.2.4. Paquetes Bluetooth

Tipos de paquetes

La especificación Bluetooth 1.1 define los siguientes tipos de paquetes para cada tipo de enlace:

- **Enlace SCO:** Se especifican cuatro paquetes destinados al transporte de voz:
 - HV1, HV2, HV3: Tienen una capacidad de 10, 20 y 30 octetos de voz, respectivamente. En el caso de los paquetes de tipo HV1, la información se protege con codificación FEC $1/3$ (*Forward Error Correction*), y con FEC $2/3$, en el caso de HV2, mientras que en los paquetes de tipo HV3 no se utiliza codificación alguna.
 - DV: Permite el envío de 10 octetos de voz y de hasta 9 octetos de datos, éstos últimos codificados con FEC $2/3$.
- **Enlace ACL:** Se especifican un total de siete paquetes: DM1, DH1, DM3, DH3, DM5, DH5 y AUX1. Salvo para este último paquete, para los de tipo DH (*Data High-Rate* o tasa de transferencia alta) y DM (*Data Medium-Rate* o tasa de transferencia media) se proporciona, como mecanismo de control de errores, la inclusión de un código de comprobación CRC. En los paquetes DM se utiliza, además, codificación FEC $2/3$. A modo de resumen, en la tabla 2.3, para cada tipo de paquete ACL, se indica el rango permitido para el tamaño del campo de datos (*payload body*) y los mecanismos de protección utilizados en cada caso.

Adicionalmente, existen paquetes de control comunes a ambos tipos de enlaces:

- *Paquete FHS:* Paquete de formato específico utilizado en los procedimientos de INQUIRY y de PAGE.

Tabla 2.3.: Tipos de paquetes ACL

Tipo de paquete ACL	Tamaño del campo de datos	FEC	CRC
DM1	0–17 octetos	$2/3$	SÍ
DH1	0–27 octetos	NO	SÍ
DM3	0–121 octetos	$2/3$	SÍ
DH3	0–183 octetos	NO	SÍ
DM5	0–224 octetos	$2/3$	SÍ
DH5	0–339 octetos	NO	SÍ
AUX1	0–29 octetos	NO	NO

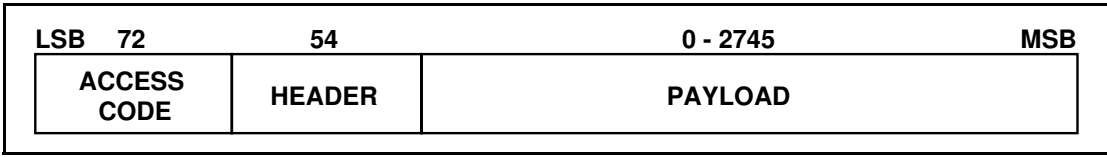
- *Paquete NULL*: Empleado para transmitir información de control de errores y de flujo relativa a tráfico ACL. No tiene que ser confirmado.
- *Paquete POLL*: Transmitido por el maestro de la *piconet* cuando, habiendo transcurrido el intervalo de *poll* —*poll interval*—, ha de direccionar a un esclavo para el que no tiene datos pendientes de envío. El esclavo que recibe un paquete POLL está obligado a confirmar al maestro su recepción.
- *Paquete DM1*: Este paquete se considera un paquete de control ya que, además de utilizarse para el envío de datos de usuario, es el paquete destinado al transporte de información de control de cualquier tipo de enlace. De hecho, tiene prioridad sobre el resto de paquetes, pudiendo incluso interrumpir la transmisión de información síncrona.

Por último, además de los paquetes de control comunes a ambos enlaces, existe un quinto paquete de control, el paquete ID, que consta únicamente del código de acceso (DAC o IAC), y es empleado en los procedimientos de INQUIRY y de PAGE.

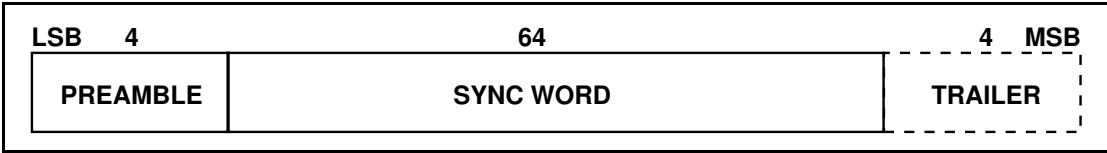
Estructura de los paquetes BB

Un paquete consta de tres campos, integrados por el número de bits que se indica en la figura 2.4a¹:

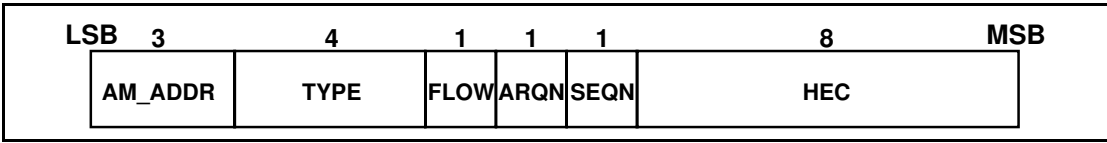
¹LSB: Least Significant Bit (Bit menos significativo). MSB: Most Significant Bit (Bit más significativo).



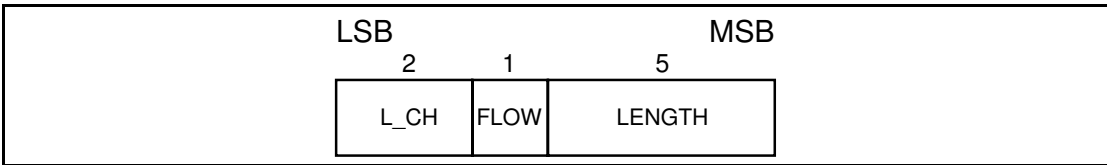
(a) Estructura del paquete



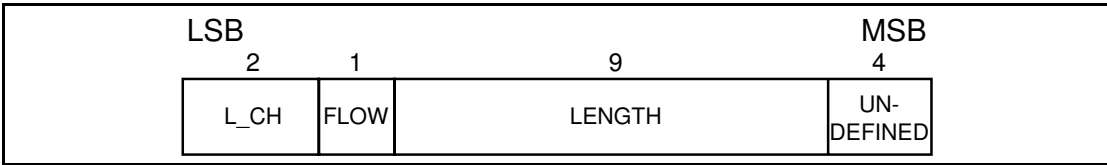
(b) Estructura del Código de Acceso (*Access Code*) del paquete Bluetooth



(c) Estructura de la cabecera (*Header*) del paquete Bluetooth



(d) Estructura de la cabecera del *payload* para paquetes de 1 *slot*



(e) Estructura de la cabecera del *payload* para paquetes *multi-slot*

Figura 2.4.: Estructura del paquete Bluetooth 1.1; Formatos de las cabeceras del paquete y del *payload*. (fuente: [Bluetooth SIG01, parte B])

- **Código de Acceso** (*Access Code*): Su estructura general se muestra en la figura 2.4b, aunque puede variar dependiendo del tipo de código. Existen tres tipos:
 - CAC (*Channel Access Code*): Permite identificar la *piconet* en la que se está intercambiando el paquete (ACL o SCO). Consta de tres campos, que suman un total de 72 bits: preámbulo (*preamble*), palabra de sincronización (*sync word*) y cola (*trailer*).
 - DAC (*Device Access Code*): Se utiliza en el procedimiento de establecimiento de conexión.
 - IAC (*Inquiry Access Code*): Se emplea en el procedimiento de búsqueda de dispositivos.

En los códigos DAC e IAC, que constituyen el único campo de un paquete ID, se omite el último campo (cola) lo que supone una extensión de 68 bits.

- **Cabecera** (*Header*): Transporta 18 bits de información de control específica del nivel BB, que se transmiten protegidos con codificación FEC $1/3$, lo que supone un total de 54 bits. Consta de los siguientes campos:
 - AM_ADDR (3 bits): Dirección del esclavo al que va dirigido el paquete.
 - TYPE (4 bits): Código de identificación del tipo de paquete.
 - FLOW (1 bit): Bit utilizado por el extremo de recepción para controlar el flujo de paquetes en el enlace ACL.
 - ARQN (*Automatic Repeat reQuest Number*) (1 bit): Bit empleado por el nivel BB del extremo de recepción para confirmar la correcta recepción del paquete recibido o, en caso contrario, para solicitar la retransmisión.
 - SEQN (*Sequential Numbering scheme*) (1 bit): Mediante este bit, el nivel BB indica si el paquete en cuestión corresponde o no a una retransmisión.
 - HEC (*Header Error Check*) (8 bits): Código de verificación de errores de la cabecera.

- **Contenido del paquete BB (payload):** Este campo encapsula los datos de control o de usuario de las capas superiores. En el caso de los paquetes ACL recibe la denominación de campo de datos (*Data Field*) y su estructura consta de:
 - Cabecera (*payload header*), de 1 ó 2 octetos, dependiendo de que el paquete ocupe un único *slot* o más, de acuerdo con el formato de las figuras 2.4d y 2.4e. Este campo contiene la longitud del campo de datos (LENGTH), el canal lógico (L_CH, *Logical Channel*) y un bit de control de flujo (FLOW). El canal lógico indica el tipo de información incluida en el campo de datos: información del nivel LM (L_CH=11) o un fragmento, inicial (L_CH=10) o intermedio (L_CH=01), de un mensaje L2CAP. Al respecto, es conveniente precisar que la transmisión de mensajes del nivel LM tiene prioridad sobre el tráfico de la capa superior L2CAP.
 - Campo de datos propiamente dicho (*payload body*).
 - Código CRC (*Cyclic Redundancy Check*): Código de verificación de errores de los datos, que no se incluye en el paquete AUX1.

2.1.2.5. Técnicas de corrección de errores empleadas

Para mejorar la fiabilidad de las transmisiones, es decir, para minimizar el número de retransmisiones, el nivel BB incorpora dos mecanismos de detección y corrección de errores a nivel de bit:

- FEC ¹/₃: Utilizado para proteger los bits de la cabecera del paquete y del campo de voz del paquete HV1, consiste en transmitir por triplicado cada bit de información. Se garantiza así la detección de 2 bits erróneos y la corrección de 1 bit.
- FEC ²/₃: Se emplea para proteger la información del campo de datos (*Data Field*) en los paquetes DV y DM, permitiendo corregir un error y detectar dos errores en cada bloque de 15 bits. Para ello, utiliza un código de Hamming (15,10) recortado (*shortened*)², lo que significa que se añaden 5 bits de redundancia por cada 10 bits de información.

²Un código (n,k) se considera recortado cuando se elimina un símbolo de información de las palabras del código, resultando ($n,k-1$).

Además, para corregir las transmisiones de paquetes DH, DM o DV erróneos, se establece un mecanismo de retransmisión basado en un esquema ARQ (*Automatic Repeat reQuest*) no numerado. El funcionamiento es el siguiente: el nivel BB del dispositivo que recibe un paquete de cualquiera de estos tipos debe confirmarlo al extremo origen en su próxima transmisión, bien positivamente (ACK, *Acknowledge*), o bien, negativamente (NAK, *Negative Acknowledge*), activando o desactivando el bit ARQN (ARQN=1 ó ARQN=0), respectivamente. Si el dispositivo receptor actúa como esclavo, está obligado a confirmarlo en el *slot* siguiente a la transmisión del maestro, mientras que en el caso de que sea el maestro, el receptor puede posponer la confirmación hasta la siguiente ocasión en la que direcciona al esclavo origen. La confirmación positiva sólo tendrá lugar en el caso de que no falle la verificación del HEC y del CRC. En caso contrario, el paquete será retransmitido hasta que se reciba una confirmación positiva o hasta que se exceda un temporizador denominado *flushTO* (véase el apartado 2.1.3).

2.1.3. Nivel LM (*Link Manager*)

El protocolo LMP (*Link Manager Protocol*) especifica todas las transacciones que pueden tener lugar entre niveles LM destinadas al establecimiento, liberación y gestión de las conexiones. En relación con esta función, las transacciones definidas pueden agruparse, grosso modo, en las siguientes categorías:

- Negociación de los cambios de modo de funcionamiento.
- Control de potencia.
- Gestión de procedimientos de seguridad: generación de la clave de enlace, activación y desactivación del cifrado.
- Gestión de los paquetes a utilizar, DM o DH.
- Gestión de la calidad de servicio.

En lo que respecta a la gestión de la calidad de servicio, concretamente, se negocia el intervalo de *poll* o T_{poll} , que se define como el máximo tiempo que puede transcurrir entre

transmisiones sucesivas del maestro a un determinado esclavo en un enlace ACL, cuando este se encuentra en modo activo. El valor mínimo definido para este parámetro, expresado como múltiplos de un *slot* y notado como n_{poll} , es de 2 *slots* (1,25 ms) y el valor por defecto es de 40 *slots* (25 ms).

2.1.4. Nivel HCI (*Host Controller Interface*)

El nivel HCI proporciona al *Bluetooth Host* una interfaz de comandos para acceder a los servicios de los niveles BB y LM, así como a los registros de estado y control del *hardware*, en definitiva, a todas las capacidades del Controlador Bluetooth (*Bluetooth Host Controller*). Para tener en cuenta que los subsistemas *Bluetooth Host* y *Bluetooth Host Controller* pueden estar físicamente separados, se ha previsto que el nivel HCI esté subdividido en tres componentes: *HCI Driver*, *HCTL* y *HCI Firmware*, de forma que el *HCI Driver* y el *HCI Firmware* se integran, respectivamente, en el *Bluetooth Host* y en el *Bluetooth Host Controller*, siendo la capa intermedia, *HCTL*, la encargada de la transferencia física de los datos entre ambos. Con este fin, la especificación Bluetooth ha definido la capa *HCTL* para SD (*Secure Digital*), UART-RS-232-C, *Three-wire UART* (3-wire) y USB (*Universal Serial Bus*) [Bluetooth SIG04, Bluetooth SIG07, vol. 4].

El *HCI driver*, en el *Host*, envía comandos —llamados comandos HCI— al *HCI firmware* del *hardware* Bluetooth con el objeto de desencadenar el inicio de una transacción del nivel LM o, alternativamente, de iniciar algún procedimiento del nivel BB o de modificar algún parámetro de dicho nivel. Asimismo, el *HCI driver* recibe notificaciones asíncronas —eventos HCI, generados por el *HCI firmware*— de los sucesos detectados por los niveles BB o LM, por ejemplo, la pérdida de un enlace.

Además, los datos —ACL y SCO— que se intercambian entre niveles homólogos de la arquitectura Bluetooth, se transfieren entre el *HCI firmware* y el *HCI driver* en paquetes específicos cuyo formato también se detalla en la parte H:1 de la especificación [Bluetooth SIG01].

2.1.5. Nivel L2CAP (*Logical Link Control and Adaptation Protocol*)

Los servicios que el nivel L2CAP proporciona a las capas superiores son los siguientes:

- Multiplexación de protocolos RFCOMM (*Serial Cable Emulation Protocol*, basado en la norma ETSI TS 07.10), SDP (*Service Discovery Protocol*) y TCS (*Telephony Control Protocol Specification*) sobre el nivel BB .
- Segmentación y ensamblado (SAR, *Segmentation And Reassembling*) de datos de niveles superiores, función imprescindible cuando los protocolos de nivel superior utilizan paquetes de longitud mayor a la manejable por el nivel BB.
- Gestión de grupos: Muchos protocolos incluyen el concepto de «grupo de direcciones». El concepto de *Grupo L2CAP* es una abstracción que permite proyectar o «mapear» eficientemente los grupos de protocolos de nivel superior en *piconets*.
- Gestión de calidad de servicio (QoS, *Quality Of Service*): L2CAP debe permitir el intercambio de información entre dos entidades L2CAP satisfaciendo los requisitos de calidad negociados.

2.2. Bluetooth 1.2, 2.0 + EDR y 2.1 + EDR

En la especificación Bluetooth 1.2, publicada en el año 2003 [Bluetooth SIG03], las mejoras introducidas respecto a la versión Bluetooth 1.1 [Bluetooth SIG01] afectan a las siguientes áreas:

- Descripción de la arquitectura.
- Establecimiento de conexiones más rápido.
- Secuencia de salto de frecuencia adaptada: AFH (*Adaptive Frequency Hopping*).
- Enlaces eSCO (*Extended SCO*).

- Inclusión de modos de operación en L2CAP con control de flujo y detección de errores.
- Capacidad de sincronización mejorada.
- Especificación de flujo mejorada.

Además, la especificación Bluetooth 1.2 fue reestructurada de forma significativa para mejorar la consistencia y legibilidad. Los cambios estructurales más importantes se llevaron a cabo en los niveles BB, LM, HCI y L2CAP. El texto en estas secciones fue reordenado, entre otras razones, para alinear las especificaciones con la nueva arquitectura y terminología. Por este motivo, en el apartado 2.2.1 se comienza presentando la arquitectura de protocolos especificada en Bluetooth 1.2. A continuación, en el apartado 2.2.2 se explica la diferencia existente entre la especificación Bluetooth 1.1 y Bluetooth 1.2 en relación con los enlaces físicos, y se describe la arquitectura definida en Bluetooth 1.2 para el transporte de datos.

Es conveniente aclarar que tanto la arquitectura de protocolos como la arquitectura de transporte de datos se han mantenido en versiones posteriores, ya que la especificación Bluetooth 2.0 + EDR es una extensión de la versión 1.2. Concretamente, la especificación Bluetooth 2.0 + EDR (*Enhanced Data Rate*) [Bluetooth SIG04], publicada en el año 2004, define los modos de funcionamiento siguientes:

- *Basic Rate* (obligatoria), correspondiente a una tasa de transferencia de 1 Mbps y compatible con versiones anteriores.
- *Enhanced Data Rate* (opcional), que utiliza una modulación PSK (*Phase Shift Keying*) y tiene dos variantes: $\pi/4$ DQPSK (*Differential Quaternary PSK*) y 8 DPSK (*Differential PSK*), que permiten conseguir, respectivamente, tasas de 2 Mbps y 3 Mbps.

En el modo de operación básico (*Basic Rate*) se engloban los siete tipos de paquetes especificados en las versiones anteriores: DM1, DH1, DM3, DH3, DM5, DH5 y AUX1. En el modo de operación ampliado (*Enhanced Data Rate*) se definen seis paquetes adicionales: 2-DH1, 3-DH1, 2-DH3, 3-DH3, 2-DH5 y 3-DH5, con una estructura distinta a la utilizada para el modo de operación básico, según se detalla en la sección 2.2.3.

En Noviembre de 2007 se hizo pública la siguiente versión, Bluetooth 2.1 + EDR [Bluetooth SIG07], que introdujo mejoras en las siguientes áreas:

- Informe de datos erróneos —*Erroneous Data Reporting*—, exclusivo de conexiones SCO y eSCO.
- Extensión del subestado *inquiry response*, consistente en contemplar la posibilidad de transmitir información sobre el dispositivo local al extremo remoto durante este subestado, utilizando un paquete de respuesta de tipo DM o DH básico.
- Evento de notificación de cambio en el temporizador de supervisión del enlace.
- Modificación del significado de uno de los campos del paquete utilizado para la transferencia de datos ACL entre el *Host* y el *Bluetooth Controller*. Concretamente, los cambios afectan al campo «*Packet Boundary Flag*» que indicaba en versiones anteriores si el paquete contenía un fragmento inicial o de continuación de protocolos de nivel superior, extendiendo su significado para permitir identificar aquellos paquetes que no deben ser descartados de forma automática.
- Extensión del modo SNIFF —SNIFF SUBRATING— para optimizar la capacidad de ahorro energético del modo SNIFF.
- Inclusión de nuevas prestaciones de seguridad: pausa y reanudación del cifrado, «*Secure Simple Pairing*» y «*Security Mode 4*».

Como puede observarse, ninguna de estas ampliaciones guarda relación con los paquetes empleados en el modo EDR o con el esquema de transmisión, por lo que el modelo que se desarrolla en la presente tesis es válido tanto para Bluetooth 2.0 + EDR como Bluetooth 2.1 + EDR, según puede comprobarse en el capítulo 5.

2.2.1. Arquitectura de protocolos Bluetooth 1.2

La figura 2.5 muestra las cuatro capas inferiores, cada una con su protocolo de comunicación asociado.

Respecto a la arquitectura definida en Bluetooth 1.1 y presentada en el apartado 2.1, puede observarse que la principal innovación introducida consiste en haber estructurado en componentes tanto los niveles que integran el subsistema *Bluetooth Controller* como el *Host*.

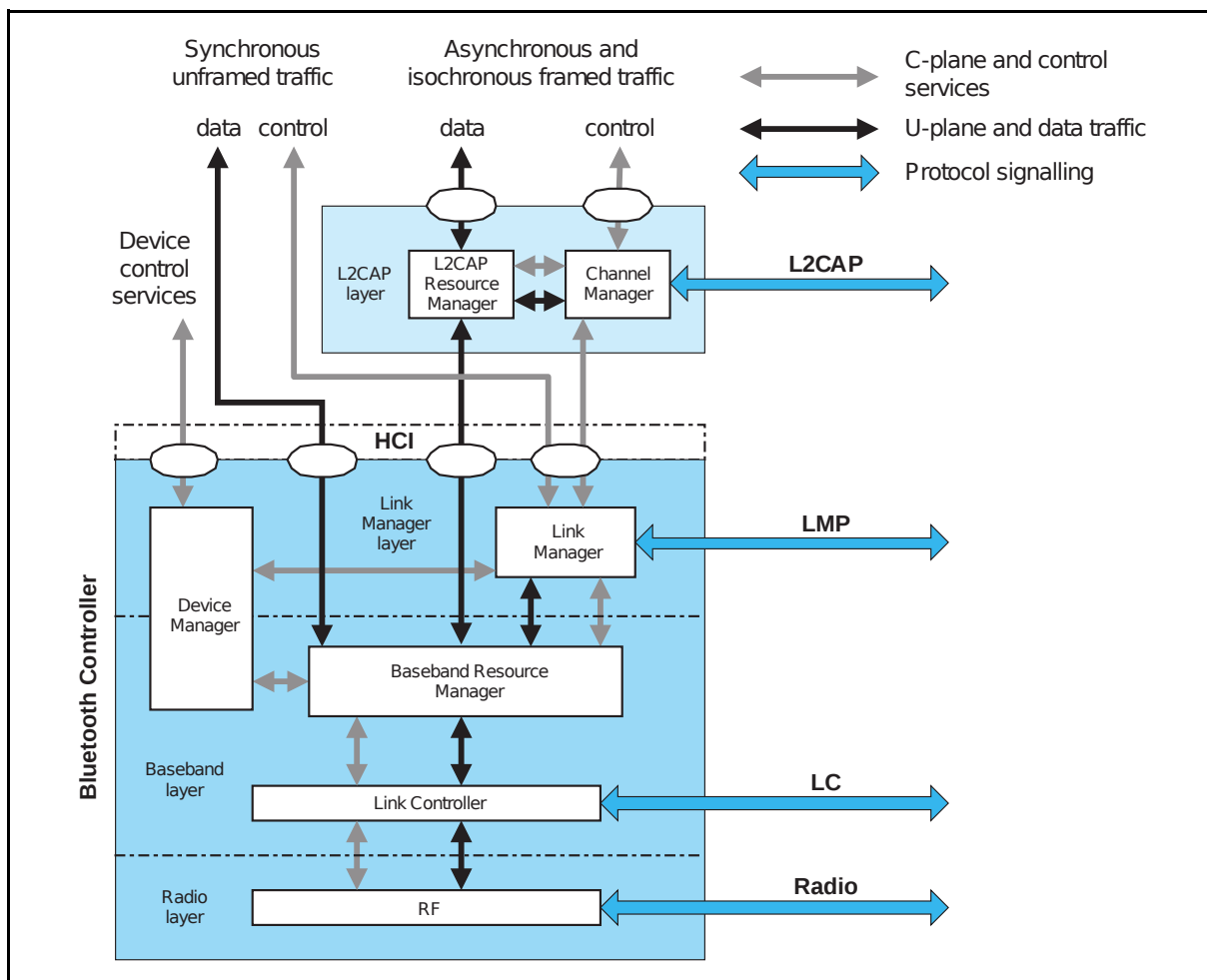


Figura 2.5.: Arquitectura de protocolos Bluetooth (fuente: [Bluetooth SIG03, Bluetooth SIG04, Bluetooth SIG07, vol. 1, parte A])

En los siguientes apartados se describen brevemente las entidades diferenciadas en cada uno de los niveles de dicha arquitectura.

Es conveniente aclarar que se profundiza más en el nivel L2CAP porque es esta capa la que proporciona nuevas funciones de control de flujo y detección de errores, y la responsable del control de calidad basado en una «especificación de flujo» (*flow specification*) similar a la definida en RFC 1363 [IETF-NWG92].

2.2.1.1. Nivel RF

Al igual que en la primera versión, el nivel RF es responsable de la transmisión y recepción de paquetes de información sobre el canal físico. La temporización y frecuencia portadora es controlada por el nivel BB. Cabe mencionar que a partir de la versión Bluetooth 1.2 el transceptor radio opera ya únicamente en la banda de frecuencia 2,4400–2,4835 GHz para la que, análogamente a Bluetooth 1.1, se definen 79 canales, espaciados 1 MHz, a las frecuencias $f = 2402 + k$ MHz, $k = 0, \dots, 78$.

2.2.1.2. Nivel BB

Se distinguen tres entidades en el nivel BB:

- *Device manager*: Es el bloque funcional del BB que controla el comportamiento general del dispositivo Bluetooth. Es responsable de todas aquellas funciones que no están directamente relacionadas con el transporte de datos, tales como la búsqueda de dispositivos o el establecimiento de conexiones. Para llevar a cabo estas funciones, solicita al subsistema de gestión de recursos (*baseband resource controller*), acceso al medio radio. También gestiona las tareas asociadas a un subconjunto de comandos HCI, como los de configuración y obtención de parámetros del dispositivo y de almacenamiento de claves.
- *Baseband resource manager*: Es responsable del acceso al medio radio y desempeña dos funciones principales: negociar los parámetros de calidad de servicio con las entidades que van a hacer uso del medio físico, y llevar a cabo una planificación de las comunicaciones y de las tareas a realizar con el fin de garantizar los criterios de calidad de servicio acordados.
- *Link controller*: A este subsistema corresponde la codificación y decodificación de paquetes Bluetooth. También lleva a cabo la señalización del protocolo relativa al control de flujo y retransmisiones.

Además, a partir de la versión Bluetooth 1.2 se modifica el diagrama de estados definido en Bluetooth 1.1 convirtiendo el modo PARK de Bluetooth 1.1 (en el que formaba parte del

estado CONNECTION) en un estado claramente diferenciado del estado CONNECTION, tal y como se ilustra en la figura 2.6.

En cuanto a la máquina de estados del nivel BB, cabe señalar que en la versión Bluetooth 2.1 + EDR se añade el modo de funcionamiento SNIFF SUBRATING que tiene como finalidad maximizar el ahorro energético del modo SNIFF.

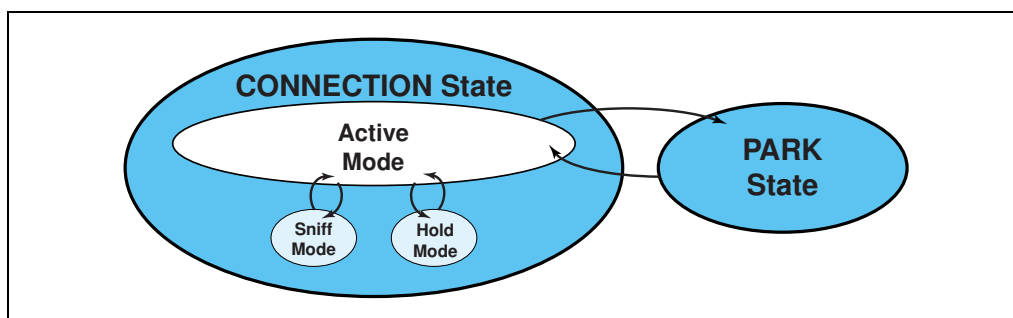


Figura 2.6.: Modificación de los estados del nivel BB a partir de la versión Bluetooth 1.2 (fuente: [Bluetooth SIG03, Bluetooth SIG04, Bluetooth SIG07, vol. 2, parte B])

2.2.1.3. Nivel LM

El *Link Manager* es responsable de la creación, modificación y liberación de enlaces lógicos (y si se requiere, de los transportes lógicos asociados), así como de la actualización de parámetros relativos a los enlaces físicos entre dispositivos. Para llevar a cabo estas funciones se comunica con la entidad homóloga del dispositivo remoto usando el protocolo LMP.

2.2.1.4. Nivel L2CAP

En Bluetooth 1.2, así como en versiones posteriores, se distinguen las siguientes entidades en el nivel L2CAP, de acuerdo con la arquitectura que se muestra en la figura 2.7:

- **Gestor de canales:** A esta entidad le corresponde la creación, gestión y destrucción de canales L2CAP para el transporte de protocolos de servicio o datos de aplicación. Centraliza la función de señalización interna: señalización para la comunicación entre

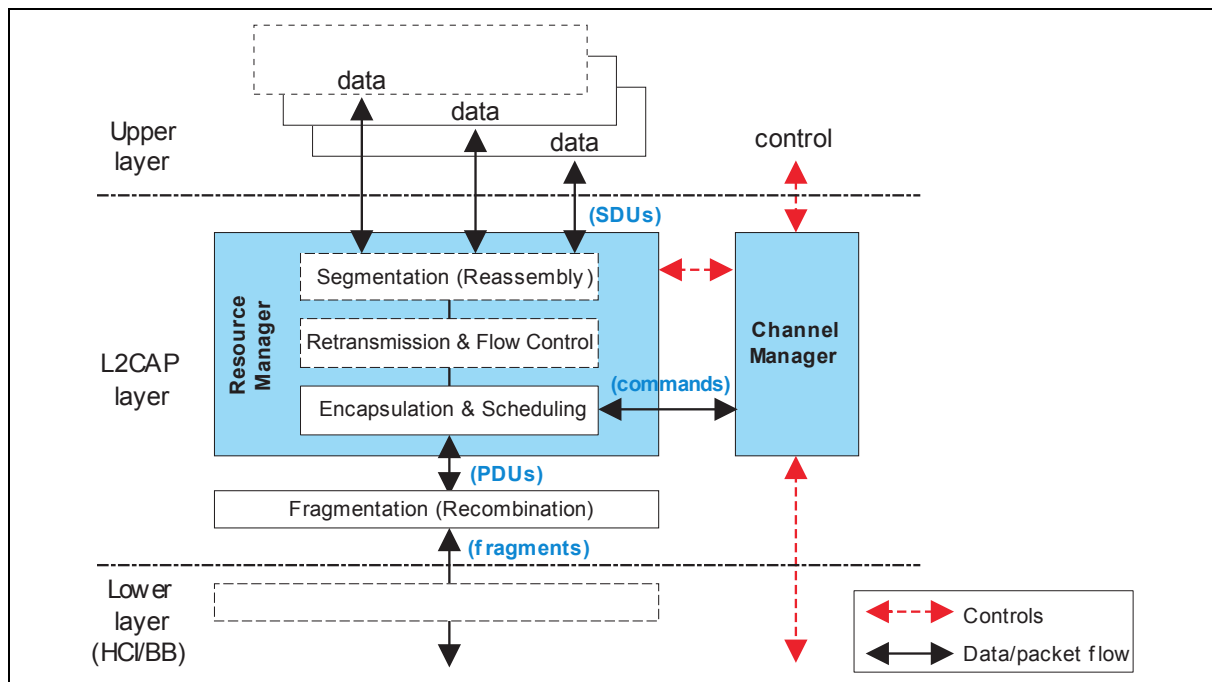


Figura 2.7.: Entidades del nivel L2CAP definidas a partir de Bluetooth 1.2 (fuente: [Bluetooth SIG03, Bluetooth SIG04, Bluetooth SIG07, vol. 3, parte A]).

entidades homólogas de L2CAP, y señalización para la interacción con los niveles superior e inferior. Lleva a cabo la multiplexación de protocolos: RFCOMM, SDP, TCS y BNEP (*Bluetooth Network Encapsulation Protocol*); e interactúa con el nivel LM local para crear nuevos enlaces lógicos (si es necesario) y para configurarlos a fin de proporcionar la calidad de servicio requerida por el tipo de datos transportado.

- **Control de recursos:** Es responsable de gestionar una entrega ordenada de fragmentos de PDU (*Protocol Data Unit*) al nivel BB, y de realizar la planificación de canales L2CAP de forma que a aquellos canales con restricciones de calidad de servicio no se les niegue el acceso al canal físico por agotamiento de recursos del *Bluetooth Controller*. Se requiere esta funcionalidad porque el modelo de arquitectura no asume que el *Bluetooth Controller* tiene *buffers* ilimitados, o que el HCI es un canal de transporte con ancho de banda infinito. Lleva a cabo las siguientes funciones:

- Segmentación y ensamblado.

- Retransmisiones y control de flujo por cada canal. Son servicios opcionales que se proporcionan a aquellas aplicaciones que requieran una tasa de error residual mucho más pequeña que la que, en principio, podría proporcionar el nivel BB.
- Encapsulación y planificación (*scheduling*).

Además, también podría supervisar que los usuarios del servicio L2CAP generan L2CAP SDUs (*Signalling Data Unit*) de acuerdo con los parámetros de calidad negociados. No obstante, se asume que las aplicaciones tienen un buen comportamiento, por lo que no se define cómo una implementación podría tratar este problema.

Adicionalmente, se detallan las ventajas derivadas de la segmentación y ensamblado, que se reproducen a continuación:

- La segmentación permite el entrelazado de los datos de las entidades de nivel superior con la finalidad de satisfacer los requisitos de latencia.
 - La gestión de memoria y de los *buffers* es más sencilla cuando L2CAP controla el tamaño de paquete.
 - La corrección de errores mediante retransmisiones es más eficiente.
 - La cantidad de información perdida cuando una PDU de L2CAP no se entrega en destino o se recibe con errores irreversibles es menor que si se perdiera la unidad de datos del nivel superior.
 - La aplicación se abstrae de la segmentación requerida para transportar sus datos en paquetes del nivel inferior.
- **Unidad de fragmentación y recombinación:** Contempla el hecho de que los niveles inferiores tengan capacidades de transmisión limitadas y puedan requerir fragmentar a tamaños distintos de los generados en la segmentación L2CAP. Esta función es de especial interés cuando el HCI está presente; en este caso, L2CAP tiene que fragmentar la L2CAP SDU teniendo en cuenta el espacio de los *buffers* del nivel BB y entregando al BB fragmentos sólo cuando esté permitido.

En definitiva, a las funciones de L2CAP especificadas en Bluetooth v1.1, que se recuerdan que son:

- Multiplexación de protocolos
- Segmentación y ensamblado
- Gestión de la calidad de servicio
- Gestión de Grupos

en Bluetooth v1.2 se añaden las siguientes:

- Control de flujo
- Control de errores y retransmisiones
- Fragmentación y recombinación

Además, a partir de Bluetooth v1.2, se contemplan tres modos de operación diferentes para L2CAP:

1. *Modo básico de L2CAP*, que es el modo por defecto. Es equivalente al funcionamiento de L2CAP en Bluetooth v1.1.
2. *Modo con control de flujo*, no se realizan retransmisiones pero se detectan las PDUs perdidas.
3. *Modo con retransmisiones*. El esquema de control de flujo está basado en la técnica de enventanado. Se utiliza un temporizador para garantizar que todas las PDUs son entregadas a la entidad homóloga.

En los modos (2) y (3) las PDUs intercambiadas entre entidades homólogas llevan un número de secuencia y son confirmadas. Estos números de secuencia se utilizan para control de *buffering* y el tamaño de la ventana de transmisión se emplea, además de para proporcionar control de flujo, para limitar el espacio de *buffer* requerido. El tamaño de la ventana junto con el parámetro *Token Bucket Size* puede utilizarse para dimensionar los *buffers*, particularmente,

para los canales que no usan control de flujo y de errores. Es conveniente aclarar que, para el estudio abordado en la presente tesis, sólo se ha considerado el modo básico.

Como ya se apuntó, en lo que concierne a la gestión de la calidad de servicio, al igual que en la versión Bluetooth 1.1, los parámetros negociados en la transacción de configuración de calidad de servicio entre los niveles L2CAP (véase el apartado 5.3 de [Bluetooth SIG04, Vol. 3, parte A]) están basados en una «especificación de flujo» (*flow specification*) similar a la definida en la recomendación RFC 1363 [IETF-NWG92].

En la especificación se contemplan tres posibles tipos de servicios: «*No traffic*», «*Best effort*» y «*Guaranteed*». Sólo es obligatorio que las implementaciones del nivel L2CAP soporten servicio «*Best effort*», que es el tipo de servicio establecido por defecto, para el que no se requiere garantizar una determinada calidad.

Los parámetros de calidad de servicio, que se describen en los siguientes epígrafes, sólo son aplicables para servicios del tipo «*Best effort*» y «*Guaranteed*»:

- **Token Rate:** Este parámetro, especificado en octetos por segundo, representa la velocidad promedio de transmisión de datos. Este valor puede ser excedido de acuerdo con los parámetros *Token Bucket Size* y *Peak Bandwidth*. Es conveniente precisar que el valor negociado entre entidades homólogas de L2CAP excluye la sobrecarga (*overhead*) del protocolo L2CAP, que ha de tenerse en consideración para la configuración local.

Cuando para este parámetro se especifica el valor por defecto (0x00000000) se indica que no existe ningún tipo de preferencia, mientras que el significado del valor máximo (0xFFFFFFFF) depende del tipo de servicio:

- *Best effort*: Se solicita todo el ancho de banda disponible.
 - *Guaranteed*: Máximo ancho de banda disponible en el momento en el que se inicia la negociación.
- **Token Bucket Size:** Este parámetro, expresado en octetos, determina el tamaño límite de las ráfagas de transmisión que una aplicación puede generar, siempre sin superar el ancho de banda de pico, dado por el parámetro *Peak Bandwidth*. Al igual que en el caso anterior, el valor negociado entre entidades homólogas de L2CAP excluye la sobrecarga

del protocolo L2CAP, que sí ha de ser tenido en cuenta para la configuración local. El valor por defecto (0x00000000) indica que este parámetro es ignorado, mientras que el significado del valor máximo (0xFFFFFFFF) también depende del tipo de servicio:

- *Best-effort*: Se solicita un tamaño de *buffer* tan grande como sea posible.
 - *Guaranteed*: Se corresponde con un valor menor o igual a la MTU (*Maximun Transmission Unit*).
-
- **Peak Bandwidth**: Este parámetro, especificado en octetos por segundo, determina el límite máximo de velocidad a la cual una aplicación puede transmitir información. Esta información puede aprovecharse para permitir una mejor asignación de recursos. El valor del parámetro *Peak Bandwidth* negociado entre dos entidades L2CAP, como en los casos anteriores, excluye la sobrecarga del protocolo. El valor por defecto (0x00000000) indica que no se impone ningún tipo de preferencia sobre el máximo ancho de banda admisible. Si se utiliza el tipo de servicio «*Guaranteed*», no debe utilizarse el valor por defecto.
 - **Access Latency**: El valor de este parámetro, expresado en microsegundos, determina el retardo máximo aceptable para la transmisión de un paquete L2CAP. Esto es, indica el retardo máximo aceptable medido desde el instante en que una L2CAP SDU se recibe desde las capas superiores hasta que se comienza su transmisión por la interfaz radio. Para su correcta interpretación, hay que tener presente su correspondencia con el nivel LM, según la cual este parámetro representa el máximo retardo transcurrido desde que el primer fragmento de una L2CAP SDU es almacenado en el *buffer* del Host Controller hasta que se inicia la transmisión radio. Por tanto, para casar ambas definiciones, el valor configurado en el *Host Controller* debe ser diferente del negociado entre entidades homólogas, ya que el primero ha de incluir el retraso de encolamiento en la transmisión del nivel L2CAP. El valor por defecto (0xFFFFFFFF) indica que no debe tenerse en consideración y no debe utilizarse en el caso del tipo de servicio «*Guaranteed*».
 - **Delay Variation**: Este parámetro, dado en microsegundos, representa la diferencia entre los retardos máximos y mínimos de una L2CAP SDU transmitida entre dos

entidades L2CAP. Tiene un carácter puramente informativo y su valor por defecto (0xFFFFFFFF) indica que no debe tenerse en consideración.

Para finalizar, se ha de mencionar que existe un parámetro, denominado *flushTO*, que es utilizado para determinar el número de retransmisiones que se realizarán a nivel BB antes de descartar un mensaje de L2CAP, cuyo valor también es negociado en la transacción de configuración de calidad. Su valor por defecto (0xFFFF) indica que se realizan un número infinito de retransmisiones, es decir, las que tengan lugar antes de que la expiración del temporizador del nivel BB *supervisionTO* indique que se ha perdido el enlace [Bluetooth SIG04, Bluetooth SIG07, vol. 3, parte A].

2.2.2. Arquitectura del transporte de datos

El sistema de transporte de datos Bluetooth presentado en Bluetooth 1.2, y mantenido en Bluetooth 2.0 + EDR y 2.1 + EDR, tiene una arquitectura estructurada en capas, como la ilustrada en la figura 2.8.

Por razones de eficiencia y compatibilidad, la arquitectura de transporte Bluetooth incluye una subdivisión de la capa lógica, distinguiéndose entre enlaces lógicos y transportes lógicos. Esta subdivisión favorece la definición de un concepto general de enlace lógico, que proporciona un transporte independiente entre dos o más dispositivos. La subcapa de transporte lógico se requiere para describir la interdependencia entre algunos tipos de enlaces lógicos.

La especificación Bluetooth v1.1 describe los enlaces ACL y SCO como enlaces físicos (*Physical Link*). Sin embargo, en versiones posteriores se explica que al agregar eSCO (*Extended SCO*) y para futuras expansiones, es mejor considerarlos como distintos tipos de transporte lógico (*Logical Transport*). No obstante, se aclara que no se logra la independencia deseada debido al uso compartido de recursos tales como la *LT_ADDR* (*Logical Transport Address*), y el esquema ARQ.

En los siguientes apartados se definen cada una de las subcapas de los niveles físicos y lógicos de la arquitectura mostrada en la figura 2.8.

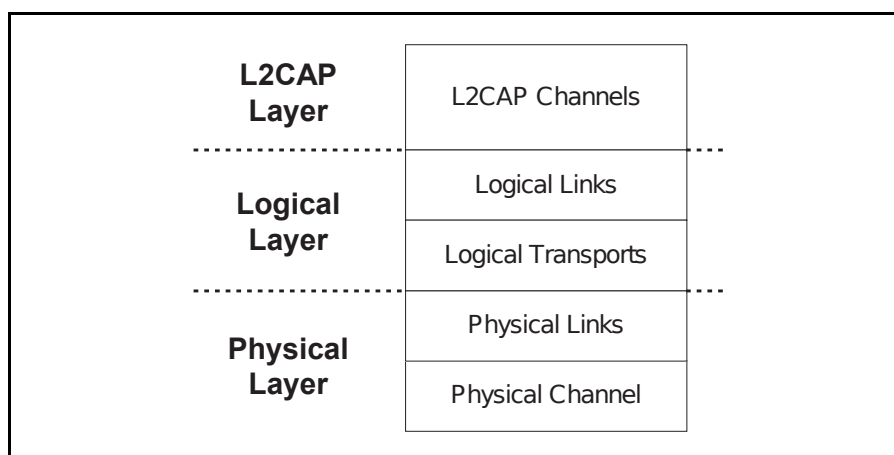


Figura 2.8.: Arquitectura del transporte de datos Bluetooth (fuente: [Bluetooth SIG03, Bluetooth SIG04, Bluetooth SIG07, vol. 1, parte A])

2.2.2.1. Canales físicos

Todos los canales físicos Bluetooth quedan determinados por la combinación de una secuencia de salto de frecuencias pseudoaleatoria, la temporización exacta del *slot* de transmisión y el código de acceso, de forma que, siempre que un dispositivo esté sincronizado en tiempo, frecuencia y código de acceso, se dice que está «conectado» a ese canal.

Se definen cuatro tipos de canales físicos, cada uno de los cuales está optimizado para un uso específico. Dos de ellos, *basic piconet channel* y *adapted piconet channel*, se utilizan para la comunicación entre dispositivos conectados y están asociados con una *piconet* determinada. Los dos restantes se emplean en los procedimientos de búsqueda de dispositivos (canal de *inquiry scan*) y de establecimiento de conexión (canal de *page scan*). Un *adapted piconet channel* se caracteriza por utilizar una secuencia de salto derivada de la secuencia de salto básica, pudiendo llegar a operar con menos de 79 frecuencias. Para ello, se requiere que los dispositivos soporten AFH.

Un dispositivo Bluetooth puede utilizar únicamente uno de los cuatro canales físicos en un momento dado. No obstante, con el fin de soportar la concurrencia de operaciones (conexión simultánea en varias *piconets*, búsqueda de dispositivos y establecimiento de nuevas conexiones), los dispositivos utilizan un esquema de multiplexación por división en el tiempo. Esto es, todos los canales físicos se subdividen en *time slots* cuya longitud dependerá

del canal físico, de forma que en un canal físico cada evento de recepción o transmisión está asociado a uno o varios *time slots*, y para cada evento de recepción o transmisión se selecciona un canal RF distinto, dado por la secuencia de salto que corresponda.

2.2.2.2. Enlaces físicos

Un enlace físico representa una conexión *Baseband* entre dos dispositivos. Un enlace físico está siempre asociado con, exactamente, un canal físico. Los enlaces físicos tienen las siguientes propiedades, que se aplican a todos los transportes lógicos definidos sobre él:

- Control de potencia.
- Supervisión del enlace (*Link Supervision*).
- Cifrado.
- Adaptación de la tasa de transmisión en función de la calidad del canal (*Channel quality-driven data rate change*).
- Control de paquetes *multi-slot*.

2.2.2.3. Transportes lógicos

Entre un maestro y un esclavo pueden establecerse cinco tipos de transportes lógicos:

1. **SCO** (*Synchronous Connection-Oriented*): El transporte lógico SCO es un enlace simétrico punto a punto entre un maestro y un esclavo concreto de la *piconet*. El maestro proporciona el sincronismo reservando *slots* periódicamente, por lo que puede considerarse como una conexión de conmutación de circuitos entre el maestro y el esclavo.

El maestro puede soportar hasta tres enlaces SCO con un mismo esclavo o con diferentes miembros de la *piconet*. Sin embargo, un esclavo puede mantener hasta tres enlaces SCO con un único maestro o dos enlaces SCO en dos *piconets* diferentes. Los paquetes utilizados en un transporte SCO, en general, nunca se retransmiten.

2. **eSCO** (*Extended Synchronous Connection-Oriented*): Es el segundo tipo de transporte lógico síncrono punto a punto entre un maestro y un esclavo determinado de la *piconet*. Puede ser simétrico o asimétrico. Al igual que en SCO, eSCO reserva *slots* y puede, por tanto, ser considerado como una conexión de conmutación de circuitos. Además, en el caso de eSCO, después de los *slots* reservados puede haber una ventana de retransmisión. Los *slots* reservados y la ventana de retransmisión constituyen lo que se denomina ventana eSCO.
3. **ACL** (*Asynchronous Connection-Oriented*): Transporte lógico punto a punto entre un maestro y un único esclavo de la *piconet*. El maestro, excluyendo los *slots* no reservados para SCO y eSCO, puede establecer un transporte lógico ACL con cualquier esclavo, incluyendo aquellos con los que ya existen SCO y eSCO.

El transporte lógico ACL proporciona una conexión de conmutación de paquetes entre el maestro y todos los esclavos de la *piconet* y soporta tanto servicios asíncronos como isócronos. Finalmente, cabe decir que entre un maestro y un esclavo puede haber únicamente un enlace ACL.

4. **ASB** (*Active Slave Broadcast*): Es utilizado por el maestro cuando tiene que transmitir a todos los esclavos activos.
5. **PSB** (*Parked Slave Broadcast*): Es empleado por el maestro para comunicar con los esclavos que se encuentran en estado *park*.

A cada esclavo en modo activo en una *piconet* se le asigna una LT_ADDR, que es una dirección de transporte lógico primaria de 3 bits, reservándose el valor nulo para los mensajes de difusión (*broadcast*). Además, se utiliza una LT_ADDR secundaria para identificar cada transporte lógico eSCO establecido en la *piconet*. Esto es, solamente tráfico eSCO (paquetes NULL, POLL y el paquetes de tipo EV negociado durante el establecimiento) puede ser enviado con la LT_ADDR secundaria. El tráfico ACL (incluyendo LMP) será transmitido con la LT_ADDR primaria. Las direcciones LT_ADDR asignadas a un esclavo sólo tendrán validez mientras que éste no se desconecte o entre en estado *park*.

2.2.2.4. Enlaces lógicos

En Bluetooth 1.2 se definen cinco enlaces lógicos:

1. **LC** (*Link Control*): El enlace lógico LC se incluye en la cabecera de todos los paquetes (con la excepción del paquete ID, que no tiene cabecera). Este enlace lógico transporta información de control lógico como ARQ, control de flujo y caracterización del *payload*.
2. **ACL-C** (*ACL Control*): El enlace lógico ACL-C, identificado por el código 11, se utiliza para el transporte de información de control intercambiada entre los niveles LM del maestro y el esclavo. Para este tipo de enlace lógico se emplean paquetes DM1.

El enlace lógico ACL-C puede ser transportado tanto sobre el transporte lógico SCO, como sobre ACL.

3. **ACL-U** (*User Asynchronous/Isochronous*): El enlace lógico ACL-U está orientado al transporte de datos de usuario asíncronos e isócronos. Estos mensajes pueden ser transmitidos en uno o más paquetes. Para los mensajes fragmentados, el paquete que contenga el primer fragmento se identifica en la cabecera del *payload* con LLID=10, y los que transporten los fragmentos de continuación, se identifican con LLID=01. En caso de no haber fragmentación, se utiliza para el LLID el valor 10.

Este enlace se transporta normalmente sobre el transporte lógico ACL. No obstante, también puede utilizarse el campo de datos del paquete DV sobre el transporte lógico SCO.

4. **SCO-S** (*User Synchronous*): El enlace lógico SCO-S se utiliza para la transferencia de datos de usuario síncronos. Este enlace lógico se transporta sobre el transporte lógico síncrono SCO.
5. **eSCO-S** (*User Extended Synchronous*): El enlace lógico eSCO-S también se utiliza para la transferencia de datos de usuario síncronos. Este enlace lógico se transporta sobre el transporte lógico síncrono eSCO.

En el caso de que los enlaces lógicos ACL-C y ACL-U compartan el mismo transporte lógico ACL, el primero deberá tener más prioridad que el segundo, excepto para las retransmisiones de paquetes ACL pendientes de confirmación, cuya prioridad deberá ser superior a la del tráfico del enlace lógico ACL-C.

2.2.2.5. Canales L2CAP

L2CAP desempeña tareas de multiplexación, permitiendo que diferentes aplicaciones compartan los recursos de un enlace lógico ACL-U entre dos dispositivos. Las entidades de nivel superior interactúan con L2CAP utilizando una interfaz orientada a canales para la creación de conexiones con entidades homólogas en otros dispositivos.

L2CAP soporta canales que pueden ser orientados a conexión u orientados a grupo. Los orientados a grupo puede mapearse sobre un enlace lógico de tipo ASB-U, o implementarse como una transmisión iterada para cada miembro utilizando enlaces lógicos ACL-U.

2.2.3. Paquetes Bluetooth a partir de Bluetooth 1.2

En este apartado se presentan los distintos tipos de paquetes especificados a partir de la versión Bluetooth 1.2 y se describe la nueva estructura del paquete Bluetooth definida por Bluetooth 2.0 + EDR.

2.2.3.1. Tipos de paquetes

A los paquetes definidos en la especificación Bluetooth 1.1, la versión Bluetooth 1.2 añade los paquetes EV (*Extended Voice*), específicos del transporte lógico eSCO, que pueden ser empleados tanto para la transmisión de voz a 64 kb/s como de datos (*transparent data*) a 64 kb/s u otras tasas. Los paquetes EV incluyen un código CRC como mecanismo de control de errores. Por tanto, son retransmitidos si su recepción no se confirma positivamente. Concretamente, se especifican tres tipos: EV3, EV4 y EV5 que transportan desde 1 octeto hasta 30, 120 y 180 octetos de voz, respectivamente, ocupando hasta 1 *slot* en el primer caso, y hasta 3 *slots* si se trata de un paquete EV4 o EV5. Sólo en el paquete EV4 la información se protege, además, con codificación FEC $2/3$. En ninguno de los casos se incluye cabecera del *payload*, de forma que la longitud el *payload* es acordada por los niveles LM durante el establecimiento de la conexión y mantenida hasta que se libera o hasta que se renegocia.

La versión Bluetooth 2.0 + EDR, para el transporte lógico eSCO, agrega los siguientes paquetes:

- Paquetes 2-EV3 y 2-EV5: Similares a los paquetes EV3 y EV5, excepto por el hecho de que el *payload* es modulado empleando $\pi/4$ DQPSK, pudiendo transportar, respectivamente, entre 1–60 octetos y 1–360 octetos de información, a los que suman 2 octetos de CRC.
- Paquetes 3-EV3 y 3-EV5: Equivalentes, también, a los paquetes EV3 y EV5; se diferencian por el hecho de que el *payload* es modulado empleando 8 DPSK, de forma que pueden enviarse entre 1–60 octetos y 1–540 octetos de información, protegidos con CRC de 16 bits.

Para el transporte lógico ACL, los paquetes ACL definidos para el envío de datos de usuario o de control se agrupan en función del modo de operación:

- Básico (*Basic Rate*): DM1, DH1, DM3, DH3, DM5, DH5 y AUX1, que son los originalmente definidos en Bluetooth 1.1 (véase el apartado 2.1.2.4).
- EDR (*Enhanced Data Rate*): 2-DH1, 3-DH1, 2-DH3, 3-DH3, 2-DH5 y 3-DH5. Se caracterizan por emplear para el campo de datos la modulación indicada:
 - $\pi/4$ DQPSK en el caso de los paquetes 2-DH1, 2-DH3 y 2-DH5, que pueden transportar, respectivamente, hasta 54, 367 y 679 octetos.
 - 8 DPSK para los paquetes 3-DH1, 3-DH3 y 3-DH5, que pueden transportar, respectivamente, hasta 83, 552 y 1021 octetos.

Tanto en el modo básico como en el modo EDR, la información se protege mediante el código CRC.

2.2.3.2. Estructura del paquete

La figura 2.9a muestra la estructura del paquete y su relación con los identificadores de los canales físicos, enlaces físicos, transportes lógicos y enlaces lógicos. Se pueden distinguir los siguientes campos:

- **Código de Acceso al Canal:** Se utiliza para identificar las comunicaciones asociadas a un canal físico determinado, y para excluir o ignorar los paquetes correspondientes a un canal físico distinto, pero que esté usando la misma portadora RF. Su formato coincide con el definido para Bluetooth 1.1 (véase la figura 2.4b del epígrafe 2.1.2.4).
- **Cabecera:** Tiene la estructura que se muestra en la figura 2.9b. Todos los campos coinciden con los definidos en Bluetooth 1.1, con la salvedad del que transporta la dirección AM_ADDR, que pasa a contener la dirección del transporte lógico (LT_ADDR).

Respecto a la identificación del transporte lógico, es conveniente aclarar que, debido a que la LT_ADDR asignada al ACL se reutiliza para el transporte lógico SCO (por razones de compatibilidad con Bluetooth v1.1), el campo LT_ADDR no es suficiente para identificar los transportes lógicos SCO y ACL, siendo necesario utilizarlo en conjunción con el tipo de paquete indicado en la cabecera. Sin embargo, el transporte lógico eSCO es identificado por una LT_ADDR específica, como ya se explicó en el apartado 2.2.2.3.

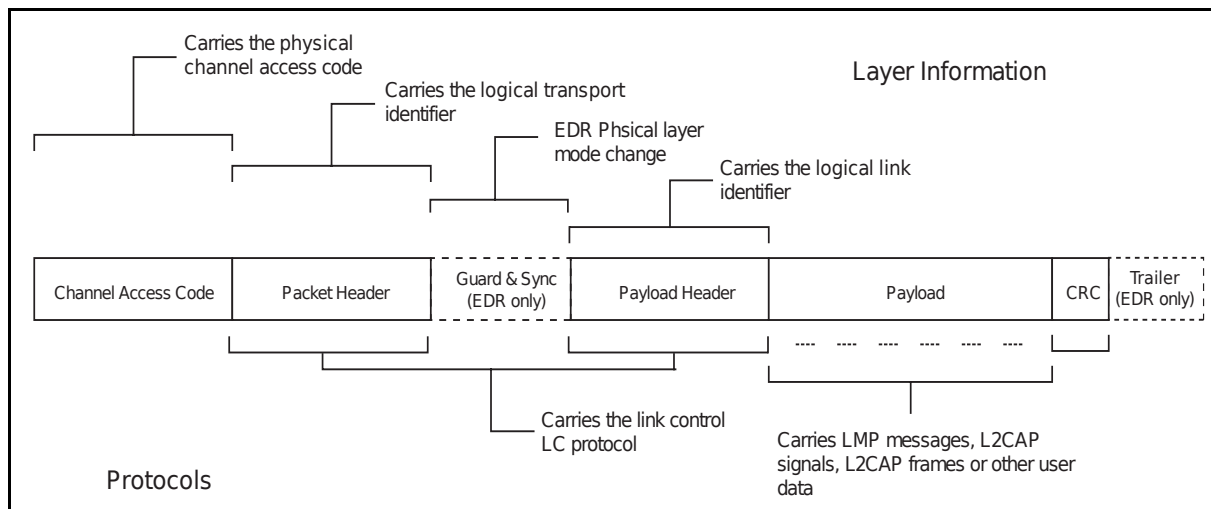
- **Payload:** Consta de los siguientes elementos: Tiempos de guarda y sincronismo EDR, cabecera del *payload*, cuerpo del *payload*, CRC y tiempo de cola EDR, que se describen a continuación:
 - Tiempo de guarda EDR (*Enhanced Data Rate Guard Time*): Tiempo entre el último símbolo GFSK de la cabecera y el primer símbolo de la secuencia de sincronización EDR, cuyo valor ha de estar comprendido entre $4,75 \mu\text{s}$ y $5,25 \mu\text{s}$.
 - Tiempo de la secuencia de sincronismo EDR (*Enhanced Data Rate Synchronization Sequence*): Consta de un símbolo de referencia DPSK (con fase arbitraria) seguido de 10 símbolos DPSK. El comienzo del símbolo de referencia debe estar sincronizado en $\pm 1/4 \mu\text{s}$ respecto al último símbolo GFSK de la cabecera, lo que supone un tiempo total de $11 \mu\text{s}$ (ya que se transmite a 1 Msímbolo/s).
 - Cabecera del *payload* (*payload header*): Consta de los siguientes campos:
 - Identificación del enlace lógico: LLID (*Logical Link Identifier*), presente en todos aquellos paquetes que se transmiten sobre transportes lógicos que

soportan enlaces lógicos ACL-U y ACL-C. Sus posibles valores, que se definen en la tabla 2.4, indican el tipo de información contenida en el cuerpo del *payload*.

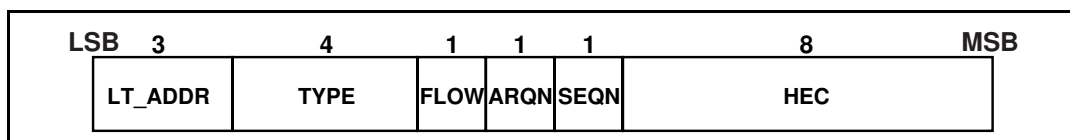
Tabla 2.4.: Significado del campo LLID de la cabecera del *payload*

LLID	Enlace Lógico	Información
00	NA	No definido
01	ACL-U	Fragmento de continuación de un mensaje L2CAP
10	ACL-U	Fragmento inicial de un mensaje L2CAP
11	ACL-C	Mensaje LMP

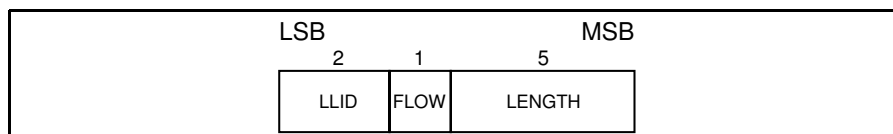
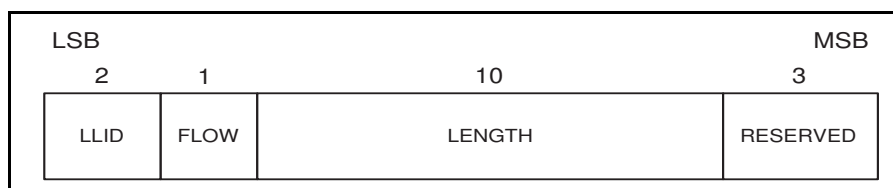
- *Flow*: Campo de 1 bit destinado a control de flujo.
- *Length*: Campo que indica la longitud del cuerpo del *payload*. Su extensión es de 2 octetos para todos los paquetes del modo EDR, y de 1 ó 2 octetos si se trata de un paquete básico, dependiendo de que ocupe un único *slot* o varios (véanse las figuras 2.9c y 2.9d).
- Cuerpo del *payload* (*payload body*): Campo de datos que transporta los datos de nivel superior: información del nivel LM, o un fragmento inicial o intermedio de una trama L2CAP.
- CRC: Código de verificación de errores de los datos, de longitud 2 octetos.
- Tiempo de cola EDR (*Enhanced Data Rate Trailer*): Para finalizar la transmisión del *payload* EDR, se agregan dos símbolos EDR correspondientes a secuencias de valor 0.



(a) Estructura general de un paquete Bluetooth



(b) Estructura de la cabecera de un paquete Bluetooth

(c) Estructura de la cabecera del *payload* para paquetes de 1 *slot* del modo básico (tipo DH1 y DM1)(d) Estructura de la cabecera del *payload* para paquetes *multi-slot* del modo básico y para todos los paquetes del modo EDR**Figura 2.9.:** Estructura del paquete Bluetooth 2.0 + EDR y 2.1 + EDR. Formatos de las cabeceras del paquete y del *payload*. Fuente: [Bluetooth SIG04, Bluetooth SIG07]

2.3. Perfiles Bluetooth

Los perfiles Bluetooth definen las posibles aplicaciones finales. Con el fin de garantizar la interoperatividad entre dispositivos de distintos fabricantes, para cada uno de los perfiles, la especificación define distintos casos de uso, detallando las características que son de obligatoria implementación para soportar un perfil y aquellas que son opcionales, los protocolos en los que se basa y los procedimientos a seguir para proporcionar un determinado servicio.

En las tablas 2.6 y 2.7 se relacionan un amplio subconjunto de todos los perfiles Bluetooth definidos.

A continuación, en los siguientes epígrafes, se introduce el perfil SPP, ya que es el que se ha elegido para extender el modelo desarrollado en [Moron08] y, también, se presenta el protocolo RFCOMM en el que se basa dicho perfil:

2.3.1. Perfil SPP y protocolo RFCOMM

El perfil de puerto serie define los mecanismos para proporcionar un servicio de emulación de una comunicación por cable serie entre dos dispositivos Bluetooth. El protocolo de transporte empleado es RFCOMM, que es un subconjunto del estándar ETSI TS 07.10 [ETSI97], concretamente, RFCOMM se restringe al modo de operación básico de TS 07.10.

RFCOMM proporciona emulación de los nueve circuitos del puerto serie RS-232 (EIA/TIA-232-E) (*Electronic Industries Alliance / Telecommunications Industry Association*), pudiendo soportar hasta 60 conexiones simultáneas entre dos dispositivos Bluetooth.

El protocolo RFCOMM se caracteriza por ser un protocolo de transporte simple, que opera sobre L2CAP y confía en que se le proporcione una transmisión fiable.

La estructura de la trama utilizada por RFCOMM, mostrada en la figura 2.10, se basa en la definida en TS 07.10, con la diferencia de que se eliminan los campos FLAG que delimitan el inicio y final de la trama. De los tipos de tramas TS 07.10 soportados por RFCOMM, que

Flag	Address	Control	Length Indicator	Information	FCS	Flag
0111 1101	1 octet	1 octet	1 or 2 octets	Unspecified length but integral number of octets	1 octet	0111 1101

Figura 2.10.: Estructura de la trama del Modo de Operación Básico del protocolo TS 07.10 (fuente: [Bluetooth SIG01, parte F:1])

se especifican en la tabla 2.5, la trama UIH es la empleada para el transporte de datos de usuario. En el campo de información (*Information*) de esta trama se pueden transmitir un número de octetos cuyo máximo viene impuesto por el parámetro *N1* (*Maximum Frame Size*). Este parámetro, que puede negociarse entre 23 y 32767 octetos utilizando la trama de tipo PN (*Parameter Negotiation*), tiene un valor por defecto de 127 octetos.

Tabla 2.5.: Tipos de tramas TS 07.10 utilizadas por RFCOMM

Tipo	Acrónimo	Significado
Comando	SABM	<i>Set Asynchronous Balanced Mode</i>
Comando	UA	<i>Unnumbered Acknowledgement</i>
Respuesta	DM	<i>Disconnected Mode</i>
Respuesta	DISC	<i>Disconnect</i>
Comando/Respuesta	UIH	<i>Unnumbered Information with Header check</i>

Tabla 2.6.: Perfiles Bluetooth (I)

Nombre del perfil	Descripción
Advanced Audio Distribution Profile (A2DP)	Describe los métodos de transmisión avanzada de audio entre dispositivos. Utiliza el protocolo AVDTP y otros perfiles como GAVDP.
Audio/Video Remote Control Profile (AVRCP)	Permite el control inalámbrico de dispositivos Audio/Video.
Basic Imaging Profile (BIP)	Proporciona mecanismos de transmisión y tratamiento básico para dispositivos de captura de imágenes.
Basic Printing Profile (BPP)	Describe un método de acceso a servicios de impresión básicos.
Common ISDN Access Profile (CIP)	Especifica el acceso inalámbrico a un dispositivo de conexión RDSI.
Cordless Telephony Profile (CTP)	Describe la comunicación entre el auricular y la base para dispositivos de telefonía inalámbrica.
Device ID Profile (DID)	Describe los métodos para permitir la identificación inalámbrica avanzada de dispositivos.
Dial-up Networking Profile (DUN)	Proporciona los mecanismos para permitir el acceso telefónico remoto a redes de datos.
Fax Profile (FAX)	Describe el método de acceso inalámbrico a dispositivos de transmisión de fax.
File Transfer Profile (FTP)	Permite la transferencia inalámbrica de ficheros entre dispositivos.
Generic Audio/Video Distribution Profile (GAVDP)	Proporciona soporte a los perfiles A2DP y VDP.
Generic Access Profile (GAP)	Describe los métodos básicos de acceso de un dispositivo Bluetooth. Es el perfil general, base para todos los demás perfiles.
Generic Object Exchange Profile (GOEP)	Proporciona los mecanismos básicos de intercambio de objetos, basado en OBEX. Sirve como base a otros perfiles más complejos.
Hard Copy Cable Replacement Profile (HCRP)	Permite el establecimiento de un enlace de datos con una impresora. A diferencia del BPP, se requieren controladores <i>software</i> para proporcionar el servicio de impresión.

Tabla 2.7.: Perfiles Bluetooth (II)

Nombre del perfil	Descripción
Hands-Free Profile (HFP)	Describe los mecanismos de acceso a un dispositivo de audio manos libres.
Human Interface Device Profile (HID)	Describe los métodos de conexión inalámbrica de dispositivos de interfaz humana como teclados, ratones, <i>joysticks</i> , etc. Está basado en el perfil USB con igual nombre.
Headset Profile (HSP)	Describe los mecanismos de acceso a un dispositivo de audio de tipo auricular.
Intercom Profile (ICP)	Describe los mecanismos de acceso a un dispositivo de audio de tipo intercomunicador.
LAN Access Profile (LAP)	Define el método de acceso a redes LAN a través del protocolo PPP.
Personal Area Networking Profile (PAN)	Describe cómo dos o más unidades Bluetooth pueden formar una red <i>ad hoc</i> y cómo acceder a redes remotas a través de puntos de acceso utilizando los procedimientos convencionales.
Phone Book Access Profile (PBAP, PBA)	Define los procedimientos para el intercambio de objetos de información telefónica entre dispositivos.
Serial Port Profile (SPP)	Define los requisitos de los dispositivos que quieran comunicarse mediante emulación de puerto serie.
Service Discovery Application Profile (SDAP)	Describe como una aplicación ha de emplear el protocolo SDP para buscar todos los servicios Bluetooth disponibles en un dispositivo remoto.
SIM Access Profile (SAP)	Permite el acceso desde un teléfono GSM, normalmente instalado en un vehículo, a la tarjeta SIM de otro teléfono de forma inalámbrica.
Synchronisation Profile (SYNC)	Este perfil, junto con el perfil GOEP, facilita la sincronización de elementos de información personal (PIM, <i>Personal Information Manager</i>), como citas o tarjetas de visita, entre dispositivos Bluetooth.
Video Distribution Profile (VDP)	Define los procedimientos para la transmisión inalámbrica de vídeo entre dispositivos.

CAPÍTULO 3

Estado de la técnica

Las tres grandes líneas de investigación que concentran el interés de la mayoría de los trabajos que abordan el estudio del rendimiento de la tecnología Bluetooth, son las siguientes:

1. Estudios de rendimiento relativos a los esquemas de transmisión TDD: En esta categoría se engloban los trabajos que versan sobre:
 - Las estrategias seguidas en *piconets* para direccionar a los esclavos.
 - La gestión del funcionamiento de *scatternets*.
 - Modelado de tráfico.
2. Estudios de rendimiento, en los que se evalúa, en función del nivel de ruido (tasa BER o relación SNR), los siguientes procedimientos de transmisión:
 - El mecanismo de retransmisión, ARQ, especificado para Bluetooth o mejoras de dicho esquema.

- El empleo de esquemas de selección dinámica de paquetes teniendo en cuenta las condiciones de canal, de cara a optimizar la eficiencia de las transmisiones e, incluso, el rendimiento de protocolos como TCP (*Transport Control Protocol*) sobre Bluetooth.
 - El uso de mecanismos de codificación alternativos.
 - La utilización de nuevos formatos de paquetes.
3. Análisis de la repercusión de la interferencia tanto de transmisiones de tecnologías que operan en la misma banda, básicamente 802.11, sobre el rendimiento de Bluetooth, como a la inversa.

Por consiguiente, la exposición del estado de la técnica se ha estructurado en torno a estas tres grandes líneas. Así, la sección 3.1 se dedica exclusivamente al estado de la técnica sobre las políticas de transmisión.

A continuación, en la sección 3.2 se recopilan los trabajos que, realizando, en la mayoría de los casos, un estudio analítico validado mediante simulación, se centran en la optimización de funciones, normalmente, pertenecientes al nivel BB, teniendo en cuenta las condiciones del canal. Además, en la sección 3.2 también se incluyen los escasos trabajos en los que se llevan a cabo medidas con dispositivos reales, encaminadas a medir, principalmente, el retardo de transmisión. En esos casos, las medidas realizadas no van precedidas de análisis teórico alguno, ni se infiere ningún modelo a partir de las medidas.

Por otro lado, en la sección 3.3, se presenta un subconjunto de los múltiples trabajos de investigación que tienen por objeto tanto el estudio de la coexistencia entre IEEE 802.11 y Bluetooth, como la propuesta de mecanismos orientados a minimizar el impacto de la interferencia provocada por una tecnología sobre la otra.

En la sección 3.4, por su parte, se incluye una síntesis de todos los trabajos que se centran en la relación existente entre la probabilidad de error y la relación SNR para las modulaciones empleadas por la tecnología Bluetooth —GFSK, $\pi/4$ DQPSK y 8 DPSK—, que han sido estudiados independientemente de que el análisis realizado se haya llevado a cabo desde una perspectiva general o no, con el fin de desarrollar, para Bluetooth 2.0 + EDR, el cálculo del retardo de transmisión teniendo en cuenta el nivel de ruido existente.

En último lugar, en la sección 3.5 se incluye una taxonomía de los artículos que se describen a lo largo del capítulo, indicando la metodología seguida y, si procede, la herramienta de simulación empleada.

3.1. Estudios de rendimiento relacionados con los esquemas de transmisión TDD

En este ámbito cabe resaltar la aportación de los autores de [Misic06], que realizan estudios del rendimiento de transmisión en Bluetooth desde el punto de vista teórico de la Ingeniería de Teletráfico. Para ello, se modelan las transmisiones entre dispositivos Bluetooth utilizando teoría de colas y cadenas de Markov. Estos estudios se realizan tanto para *piconets* Bluetooth aisladas, como para *piconets* interconectadas entre sí (*scatternets*).

Los aspectos de funcionamiento en los que se centran abarcan desde los esquemas de *poll* tradicionales [Misic03d], pasando por los que introducen mejoras como, por ejemplo, aquellos que priorizan los esclavos o consideran el efecto de la capacidad finita de los *buffers* [Misic04c] hasta, incluso, los que se centran en la utilización esquemas de *poll* adaptativos [Misic03e, Misic04e]. Los resultados obtenidos se aplican a la resolución de problemas como el control de admisión en *piconets* [Misic03a, Misic04a] y la reserva de ancho de banda para garantizar calidad de servicio [Misic04d].

Análogamente, los autores investigan las políticas de segmentación y reensamblado, aplicándolas a la transmisión de tráfico TCP/IP (*Transport Control Protocol / Internet Protocol*) [Misic05a] y, también, al envío de tráfico síncrono SCO [Misic03g].

Respecto a las *scatternets*, los mismos autores comienzan investigando el funcionamiento de los dispositivos puente (*bridge*) entre *piconets* [Misic02c], tanto maestro-esclavo [Misic02a] como esclavo-esclavo [Misic02b, Misic02d], para continuar aplicando los esquemas de *poll* investigados [Misic03c, Misic03b, Misic03f], extendiéndolos con mecanismos que permiten la sincronización entre *piconets* (*rendezvous*, es decir, mecanismos de encuentro) [Misic04b]. Finalmente, para el estudio de *scatternets* complejas, integran otros efectos anteriormente estudiados, como los *buffers* finitos [Misic05b].

En todos los casos la validación de los resultados analíticos se lleva a cabo mediante simulaciones, utilizando el simulador de redes de Petri, Artifex [RSOFT], y el simulador de redes de eventos discretos Network Simulator (NS-2) [NS-2].

Entre otros trabajos, pertenecientes a este área de investigación, que citan alguno de los estudios de los autores de [Mistic06], cabe mencionar los presentados en [Zhihong09] y [Hsu10], que proponen un algoritmo de *poll* adaptativo. En el primer caso, la adaptación se basa en la estimación de la covarianza del tráfico entre los miembros de una *piconet*. Por contra, los autores de [Hsu10] lo hacen teniendo en cuenta las distintas circunstancias del tráfico generado hacia cada esclavo para maximizar el rendimiento de transmisión. Con este fin, realizan un análisis teórico exhaustivo, comparando su propuesta con otros métodos existentes y la validan, mediante simulación, empleando también NS-2.

El trabajo presentado en [Zussman07] destaca por la evaluación que efectúa del funcionamiento del esquema TDD de Bluetooth, ya que analiza los algoritmos de *poll* convencionales con tráfico simétrico y asimétrico, proporcionando soluciones empíricas en algunos casos y facilitando, además, resultados numéricos mediante simulación con OPNET [OPNET].

En los esquemas de *poll* con prioridad se centran los trabajos presentados en [Kalvala06], [Lee07] y [Yang10]. Mientras que [Lee07] se limita a comparar varios algoritmos de *poll* basados en *Round-Robin* y a proponer un algoritmo sencillo, en [Kalvala06] el método propuesto se combina con técnicas de segmentación para mejorar las prestaciones de la *piconet*. Su efectividad en relación con otros algoritmos se evalúa utilizando un simulador de eventos discretos en C++, mismo método utilizado también por [Lee07].

Entre las propuestas de algoritmos de *poll* específicas para su funcionamiento en *scatternets* se encuentran las incluidas en [Roy07] y [Reddy08], con el matiz de que en [Roy07] el algoritmo propuesto se complementa con un algoritmo de construcción dinámica de *scatternets*. La validación de este algoritmo se lleva a cabo mediante simulación, utilizando OPNET, aunque no se indica explícitamente. Sin embargo, en [Reddy08] para la realización de simulaciones los autores indican que han empleado NS-2.

3.2. Estudios de rendimiento relacionados con procedimientos de transmisión

En [Chen04a] se propone una mejora del mecanismo ARQ para compensar la degradación del canal, a fin de soportar transmisiones de audio en tiempo real (*audio-streaming*): ARQ RTO (*Retransmission TimeOut*) adaptativo. La mejora propuesta consiste en limitar el número de retransmisiones, modificándolo de forma dinámica, en función de las condiciones del canal, estimadas a partir de la tasa BER del histórico de las transmisiones de paquetes previos. Las ventajas del algoritmo propuesto han sido evaluadas empíricamente utilizando BlueZ [BlueZ]. Para ello, los autores han modificado la implementación de los protocolos BNEP y L2CAP. Concretamente, el algoritmo extrae información de la cabecera correspondiente a BNEP del paquete RTO recibido y la transfiere al nivel subyacente L2CAP y este, a su vez, a HCI. Cada paquete RTP (*Real Time Protocol*) es dividido en múltiples fragmentos por el nivel L2CAP, de forma que una vez que el nivel HCI los recibe, los encola, almacenando el tiempo de llegada del primer fragmento y midiendo el RTT (*Round Trip Time*) para todos los fragmentos del paquete RTP correctamente transmitidos. Así, en caso de que alguno de los fragmentos del paquete RTP sea descartado debido a la expiración del temporizador del esquema ARQ RTO, los restantes fragmentos del paquete serán eliminados del *buffer* de HCI y del nivel BB empleando, en este último caso, el comando *HCI_Flush*. Cabe destacar que, alternativamente al enfoque propuesto podría utilizarse el comando *HCI_Write_Automatic_Flush_Timeout* que permite modificar, para una conexión Bluetooth, el temporizador *Flush_TO*. De este modo, no sería necesario haber modificado la implementación de la pila de protocolos BlueZ. Además, los autores no detallan el tamaño al que se lleva a cabo dicha fragmentación ni el motivo por el que la efectúan. Adicionalmente, tampoco tienen en cuenta el efecto de los *buffers* de los dispositivos. De hecho, los resultados sólo ponen de manifiesto que al descartar la información que se encuentra bloqueada en los *buffers* de entrada del dispositivo debido a las retransmisiones originadas por la interferencia en el enlace radio, se mejora el tiempo de latencia del enlace de datos.

No obstante, dejando al margen estas consideraciones, resulta de especial interés que en este estudio los autores resalten que uno de los retos que han tenido que afrontar ha sido la generación fiable de la interferencia, explicando que el problema radica en la dificultad de

controlar el nivel de señal, ya que los factores ambientales hacen que la calidad del enlace sea impredecible. La solución adoptada por los autores consiste en emplear dispositivos 802.11 para crear interferencia, al igual que se hace en [Moron08]. Además aclaran que, dado que al incrementar la distancia física entre los dispositivos disminuye la calidad del enlace, dificultándose aún más el control de las condiciones de medida, ha sido necesario mantener muy próximos los dispositivos Bluetooth y regular la carga de tráfico en las transmisiones 802.11. También, análogamente a como se procede en [Moron08], se obtiene la calidad del enlace empleando el comando *HCI_Get_Link_Quality*, valor a partir del que se determina la tasa BER mediante la transformación especificada por el fabricante CSR.

Este mismo procedimiento de obtención de la tasa BER es el que se sigue en el trabajo presentado por el mismo autor en [Chen04b]. En este caso, el objetivo es investigar el efecto sobre el rendimiento de TCP de la selección dinámica del tipo y tamaño de paquete adecuado a las condiciones del canal, en función de la tasa BER, demostrando que la tasa efectiva de transmisión de TCP puede mejorarse significativamente si se utiliza un paquete de tamaño determinado. A diferencia del caso anterior, los resultados mostrados han sido obtenidos mediante simulación utilizando el *software* NS-2 [NS-2].

Por otro lado, el análisis del impacto de la tasa BER en el rendimiento de TCP/IP, objeto de estudio de [Chen04b], ha sido abordado con anterioridad en [Chui02]. Concretamente, en [Chui02] se presenta un simulador de eventos discretos para modelar la eficiencia de una *piconet* bajo condiciones de carga y tasa BER variables, teniendo en cuenta que el rendimiento de la red depende de aspectos del nivel físico tales como el tipo de código FEC empleado y el tamaño del paquete. Las simulaciones se han llevado a cabo para una *piconet* con dos esclavos en la que el maestro actúa como un punto de acceso a Ethernet transportando tráfico TCP/IP. En las simulaciones realizadas los autores asumen, entre otros supuestos, que en la cabecera protegida con codificación FEC $1/3$ nunca se producen errores y, también, que la tasa BER es constante durante el tiempo de transmisión de cada paquete. Como resultados se obtienen gráficas de rendimiento comparando la utilización de los paquetes definidos en Bluetooth 1.1, según las condiciones de BER establecidas en la simulación. A partir de los resultados presentados los autores concluyen que cambios en la tasa BER de 10^{-6} a 10^{-4} , en caso de utilizar paquetes de 5 *slots* sin codificación FEC, causan que el retardo aumente significativamente de 53 ms a 373 ms para el 90 % de los paquetes transmitidos, mientras que en el caso de paquetes DM el tiempo de transmisión permanece constante en torno a

un valor de 77 ms. Sin embargo, el efecto de la fragmentación de las capas superiores de Bluetooth es completamente ignorado en las simulaciones, así como el funcionamiento real de una *piconet* Bluetooth. Adicionalmente, no se detallan las herramientas empleadas para la simulación o cómo han sido validadas.

Aunque no con el objetivo específico de optimizar el rendimiento de TCP, como en [Chen04b], también en [Tan09] los autores proponen un algoritmo para seleccionar dinámicamente el tipo de paquete Bluetooth óptimo, basándose en la información estadística obtenida a partir del esquema ARQ. Partiendo de resultados teóricos de trabajos de otros autores, concluyen que, asumiendo un canal determinado (AWGN o Rayleigh), el máximo rendimiento de transmisión se obtiene utilizando cuatro tipos de paquetes Bluetooth (excluyendo los definidos en el modo EDR), según el rango de la relación SNR por bit. Sin embargo, determinar este parámetro, o la tasa BER asociada, para valorar las condiciones del canal durante la transmisión con una precisión aceptable es una tarea difícil. Por tanto, los autores proponen utilizar las estadísticas acumuladas del esquema de retransmisión ARQN para ejecutar un algoritmo de selección del tipo de paquete. Mediante simulación (aunque no indican la herramienta utilizada) concluyen que se logra una mejora significativa seleccionando de forma dinámica los paquetes a utilizar respecto al caso en el que no se lleva a cabo dicha adaptación. Sin embargo, en la práctica los dispositivos reales no fijan un tipo de paquete, sino que lo adaptan según la cantidad de información a transmitir. Igualmente, la mayor parte de los dispositivos reales ya utilizan algún mecanismo de adaptación a las condiciones del canal, aunque esta información no suele ser pública.

En el trabajo presentado en [El-Bendary09b] los autores explican cómo lograr una mejora de la tasa de transmisión en un enlace Bluetooth utilizando un esquema de selección del tipo de paquete adaptativo, considerando tanto paquetes del modo básico como del modo EDR. Concretamente, se emplea CQDDR (*Channel Quality Driven Data Rate Change*) para seleccionar el tamaño del paquete teniendo en cuenta las condiciones del canal, utilizando el indicador RSSI (*Received Signal Strength Indicator*). Además, los autores, para canales AWGN y de tipo Rayleigh no selectivo en frecuencia (*Rayleigh-flat fading*), proponen formatos de paquetes alternativos en los que el número de bits de redundancia introducidos en el código de acceso y en la cabecera por los esquemas de codificación BCH (64, 30) y FEC $1/3$ especificados en el estándar, puedan reservarse para extender el *payload* y, por consiguiente, incrementar la tasa de transmisión cuando las condiciones del canal sean buenas. Para

comparar el rendimiento conseguido con las modificaciones propuestas en relación con el alcanzable utilizando el esquema de codificación empleado en Bluetooth, los autores llevan a cabo, en Matlab [MathWorks], simulaciones basadas en el método de Montecarlo. Además de que la propuesta supone una modificación de las especificaciones Bluetooth, cabe resaltar que recuperar los 70 bits de redundancia añadidos en el código de acceso y en la cabecera para aumentar la capacidad del campo de datos, no supone un incremento considerable de la longitud útil de datos, sobre todo en paquetes *multislot*. Nótese que los paquetes EDR 2-DH y 3-DH, duplican y triplican, respectivamente, las capacidades de los paquetes DH del modo básico. Por otro lado, los autores indican que la modulación empleada en el simulador es BPSK (*Binary Phase Shift Keying*). Sin embargo, al respecto hay que tener presente que puede no ser lo suficientemente realista como para evaluar la tasa de transferencia del modo EDR, en el que se emplean las modulaciones $\pi/4$ DQPSK (2 Mbps) y 8 DPSK (3 Mbps). Por último, más grave resulta el hecho de que los autores realizan sus experimentos de simulación empleando paquetes 2-DM1, tipo de paquete no definido en el estándar Bluetooth. Es decir, consideran al paquete 2-DM1, erróneamente, un paquete EDR, no proponiendo su inclusión en el estándar, propuesta que sí se realiza en [Chen06, Xu07], pese a que citan el trabajo [Chen06].

Los autores de [Feng09] también proponen un esquema adaptativo de selección del paquete a utilizar en la próxima transmisión en función de la tasa de paquetes erróneos. Para ello, comienzan estudiando el rendimiento de Bluetooth 2.0 + EDR, analizando el efecto de la calidad del enlace sobre la tasa de transferencia y llevando a cabo la simulación de un modelo de Bluetooth para Simulink de Matlab. El modelo, que incluye los niveles RF y BB, simula el establecimiento de una conexión y permite configurar parámetros como el posicionamiento, efectos de propagación y características radio. El objetivo de los autores es evaluar, mediante simulación, el rendimiento del sistema Bluetooth en términos de tasa de transferencia, utilizando distintos tipos de paquete, y simulando escenarios de coexistencia con otros dispositivos ISM o con otras *piconets* Bluetooth, para analizar el efecto de las tasas FER (*Frame Error Rate*) y BER sobre la tasa de transferencia. Sin embargo, las expresiones de la tasa BER empleadas no corresponden a las modulaciones de Bluetooth, o no se han utilizado de forma adecuada para normalizar potencia transmitida por símbolo.

En la misma línea que el estudio presentado en [Chen04a], aunque con una metodología distinta (estudio analítico en lugar de medida experimental), cabe mencionar el trabajo

[Zanella09b], en el que se propone una mejora del esquema ARQ consistente en no retransmitir un paquete con datos hasta que no se haya recibido explícitamente confirmación negativa (NAK). Esto es, suponiendo que el maestro transmite un paquete que es recibido por el esclavo, en caso de que el paquete de respuesta enviado por el esclavo contenga algún error, por ejemplo en la cabecera, no se procedería a la retransmisión inmediata de los datos pendientes de confirmación, sino a solicitar la retransmisión de la respuesta generada por el esclavo. Aunque el autor, basándose en los fundamentos analíticos desarrollados en [Zanella09a], concluye que se consigue una mejora significativa en la eficiencia energética, cabe argumentar que, tal y como se especifica en el estándar (véase el apartado 7.6.2 de [Bluetooth SIG04, Bluetooth SIG07, vol. 2, parte B]) la probabilidad de que la confirmación transmitida en el paquete de respuesta sea errónea es muy baja, dado que la cabecera del paquete está fuertemente protegida. Además, el autor no tiene en cuenta que los paquetes que el maestro tendría que utilizar para seguir esta propuesta hacen que no sea factible. Esto es, obvia el hecho de que para obligar al esclavo a responder tendría que utilizar un paquete POLL y la respuesta a dicho paquete es una confirmación implícita a la recepción de dicho paquete, no pudiendo relacionarse con transmisiones anteriores. Por otro lado, el paquete NULL tampoco sería una opción, ya que puede no ser confirmado. El estudio no realiza ninguna comprobación mediante simulaciones o mediante experimentos o implementaciones reales.

Con el objetivo de analizar el rendimiento en presencia de ruido en función de la relación SNR y teniendo en cuenta el esquema de retransmisiones, pero esta vez considerando el especificado en el estándar Bluetooth, ARQ (*Stop&Wait ARQ*), cabe también mencionar el trabajo desarrollado por el mismo autor en [Zanella09a]. Este trabajo presenta un marco matemático para el análisis del rendimiento de las conexiones Bluetooth en canales con desvanecimiento. Utiliza un modelo basado en cadenas de Markov para distinguir la transmisión inicial de tramas frente a las retransmisiones. A partir de este modelo, aplicando la teoría de la renovación (*renewal reward theory*), que es una generalización de los procesos de Poisson, obtiene expresiones del rendimiento de transmisión (retardo y tasa efectiva de transmisión) y eficiencia energética. Este modelo lo aplica a Bluetooth EDR concluyendo que los mejores resultados se obtienen utilizando paquetes de máxima longitud de 2 Mbps para SNR medias o 3 Mbps para relaciones SNR altas. Tampoco en este caso se lleva a cabo validación alguna, ni mediante simulaciones ni experimentalmente.

En el grupo de trabajos que se centran en estudios de rendimiento evaluando, en función del nivel de ruido, aspectos del protocolo relacionados con los procedimientos de transmisión, cabe mencionar aquellos que proponen el uso de mecanismos de codificación alternativos, como los Turbo-Códigos (TC, *Turbo Code*) [Valenti02a, Moon08, Kahveci10], FEC entrelazado o IFEC (*Interleaved-FEC*) [Chen04c, Chen06], y códigos convolucionales [El-Bendary09a].

Así, en [Valenti02a] se propone la utilización de Turbo-Códigos para mejorar el funcionamiento de las redes Bluetooth en entornos sometidos a fuertes interferencias. Con el objetivo de no modificar las especificaciones Bluetooth existentes, propone el uso de paquetes AUX1 para el transporte de datos codificados utilizando Turbo-Códigos de tasa compatible (*RCPT codes*). La utilización de este tipo de códigos se justifica dada la reducida longitud del paquete AUX1 respecto a la longitud de los Turbo-Códigos convencionales. La codificación y decodificación efectiva de la información debe realizarse externamente al transceptor Bluetooth y, adicionalmente, es necesario contar con información del nivel de señal recibido (RSSI) para que la decodificación se realice de manera óptima. Igualmente se debe introducir un mecanismo ARQ externo para garantizar el funcionamiento efectivo de los códigos propuestos. Los resultados expuestos parten de cálculos analíticos combinados con simulaciones (utilizando el método de Montecarlo), demostrando las ventajas obtenidas para valores de SNR bajos. Como crítica a la propuesta, cabe señalar que la utilización de los paquetes AUX1 no es evidente si se emplean módulos comerciales, ya que el estándar no proporciona un método estándar de selección de este tipo de paquete a través de la interfaz HCI. Además, para los valores de SNR en los que se presentan mejoras, el funcionamiento de un dispositivo real no sería posible ya que la información de control transmitida, al igual que el resto de los campos del paquete, no estarían protegidos. Asimismo, la complejidad involucrada en el proceso no facilita su posible incorporación a un sistema inalámbrico de bajo coste como lo es Bluetooth.

En [Moon08] se simula el rendimiento de la transmisión de paquetes ACL DM1, DM3 y DM5, sustituyendo el esquema de protección FEC $2/3$ por el empleo de Turbo-Códigos. En la transmisión de paquetes TCP, supuestamente, se ha tenido en cuenta el efecto de fragmentación sucesiva de las distintas capas del estándar Bluetooth. Sin embargo, a lo largo del artículo, no se justifican ni detallan muchos aspectos:

- No se indica la herramienta de simulación o método empleado.

- No se especifican las fórmulas empleadas para determinar la tasa BER respecto a la relación SNR correspondiente a la modulación GFSK.
- No se indica la ganancia de los Turbo-Códigos, limitándose a dar una cota general de la tasa BER para relaciones SNR elevadas. No obstante, tampoco se especifican referencias ni límites concretos de la misma.
- No realiza una comparación con la codificación estándar utilizada por Bluetooth, FEC $2/3$.
- No se indica la complejidad de la implementación, ni otros inconvenientes y, además, de los resultados no se extraen conclusiones.

En [Kahveci10] también se investiga la eficiencia de los Turbo-Códigos. En este caso, los autores emplean un algoritmo de decodificación MAP (*Maximum-A-Posteriori*) modificado para paquetes de datos Bluetooth en entornos de interior, para lo que calculan la tasa BER en función de la relación SNR. Aunque indican que para entornos de interior se emplea el modelo de canal de Rice, en su propuesta utilizan el modelo de Rayleigh al considerar que no hay visión directa. A partir de este modelo, proponen la utilización de Turbo-Códigos para mejorar el rendimiento en casos de relación SNR desfavorable. Además, los autores proponen para disminuir la complejidad de decodificación de los Turbo-Códigos emplear algoritmos iterativos MAP. Así, partiendo de la formulación matemática de los algoritmos y del modelo de canal propuesto, obtienen resultados mediante simulación. Sin embargo, en el modelo planteado para realizar la simulación suponen la utilización de la modulación binaria BPSK, pese a que Bluetooth realmente utiliza GFSK. Igualmente, aunque describen los métodos FEC utilizados en la codificación de los paquetes Bluetooth, en la presentación de resultados realizan la comparación suponiendo codificación FEC $1/3$, aunque en realidad se emplea FEC $2/3$, que según se indica en los fundamentos teóricos, se basa en un código Hamming recortado (15,10). Finalmente, cabe destacar que, paradójicamente, los resultados que obtienen con esta codificación —Hamming recortado (15,10)— son mejores que si emplean los algoritmos MAP propuestos.

En lo que respecta a la utilización de codificación IFEC para proteger las transmisiones frente a errores aleatorios y en ráfaga cabe mencionar el presentado en [Chen04c]. Los autores aclaran que el código IFEC propuesto utiliza también el código Hamming (15,10),

que es el especificado por Bluetooth para los paquetes DM, de forma que el número de bits de redundancia introducidos es el mismo. Sin embargo, según explican los autores, la codificación IFEC se caracteriza por aplicar entrelazado a nivel de bit en lugar de hacerlo a nivel de paquete. Por tanto, a la robustez que proporciona la codificación FEC frente a los errores aleatorios, se suma la resistencia a los errores que se puedan producir en ráfagas. En un trabajo publicado dos años después, [Chen06], el mismo autor extiende su empleo al modo EDR, lo que lleva implícita una propuesta adicional: la definición de paquetes 2-DM1/3/5 y 3-DM1/3/5 no incluidos en el estándar. Este es también el objetivo del trabajo presentado en [Xu07], en el cuál se propone también la introducción en el estándar Bluetooth de paquetes de alta velocidad (EDR) que hagan uso de la codificación FEC $2/3$ ya incorporada en los paquetes DM de modo básico. De esta forma se dispondrían de paquetes 2-DM1, 2-DM3, 2-DM5, 3-DM1, 3-DM3 y 3-DM5, de mayor longitud que los DM básicos, pero con una mejor robustez ante relaciones SNR bajas que los DH EDR correspondientes. Partiendo de las expresiones que calculan la tasa BER para las diferentes modulaciones, el autor deriva analíticamente la tasa efectiva de transmisión (*throughput*) para los nuevos paquetes. Finalmente, realiza una comparación en función de este parámetro con los paquetes existentes, derivando una regla de selección frente a la relación SNR presente en un canal AWGN. Sin embargo, en el desarrollo analítico emplea expresiones de BER para las modulaciones que no parecen ser correctas.

En [El-Bendary09a] los autores presentan un nuevo formato de paquete EDR que emplea esquemas de codificación diferentes: *Hamming expurgated* y códigos convolucionales. También estudian analíticamente la probabilidad de paquetes erróneos, tanto para el formato estándar como para el propuesto y considerando el modelo de canal AWGN. Mediante simulaciones de Montecarlo realizadas con Matlab, concluyen que se consigue una mejora energética al tiempo que se reduce la tasa de paquetes erróneos.

En cuanto al análisis del efecto de la tasa BER sobre el rendimiento, desde un punto de vista empírico, se encuentra el estudio presentado en [Warriach08], que pretende evaluar experimentalmente el rendimiento de las transmisiones Bluetooth entre un maestro y un esclavo considerando los siguientes parámetros: retardo de transmisión de los paquetes Bluetooth, tasa BER y tasa PLR (*Packet Loss Rate*). Cada dispositivo Bluetooth empleado en el escenario de medida descrito por los autores consta de un transceptor Bluetooth y de un microcontrolador HCS12, en el que se ha implementado la pila de protocolos Bluetooth

incluyendo el protocolo RFCOMM. Como capa de transporte HCTL para la comunicación entre el microcontrolador y el módulo Bluetooth se ha optado por emplear la interfaz UART, de forma que el estudio de rendimiento realizado se basa en la monitorización de dicha interfaz UART de ambas unidades, maestro y esclavo, utilizando el mismo instrumento de medida, un tercer microcontrolador en el que se implementa el *software* para la medición de retardo de transmisión y de estadísticas de errores de bits y paquetes perdidos. Aunque los autores explican que para evitar el efecto de las interferencias generadas por otros sistemas Bluetooth o WLAN (*Wireless Local Area Network*) las medidas se han llevado a cabo en una cámara anecoica, no se describe cómo han inducido pérdidas en el sistema de cara a medir las tasas BER y PLR. Al respecto, llama la atención que sí se especifica un valor concreto para el parámetro indicativo de la calidad del enlace, retornado por el comando *HCI_Get_Link_Quality*, pese a que no se menciona cómo se provoca la interferencia, ni el momento ni tampoco el número de veces en el que se realiza la obtención de este parámetro a lo largo de las 2000 transmisiones realizadas. Adicionalmente, los autores apuntan que obtienen el valor del parámetro RSSI a partir del evento *HCI_Inquiry_Result_with_RSSI_Event*, acción que carece de sentido dado que las medidas objeto de análisis conciernen a la transmisión de datos. La razón que cabe argumentar es que, para medir el parámetro en cuestión, el estándar Bluetooth contempla un comando específico *HCI_Read_RSSI*, no vinculado a la realización de un procedimiento de INQUIRY. Esto es, el evento *HCI_Inquiry_Result_with_RSSI_Event* utilizado por los autores retorna el parámetro RSSI medido durante la fase de respuesta de un dispositivo a un mensaje de INQUIRY (subestado INQUIRY_RESPONSE) y es generado, únicamente, en respuesta a un comando *HCI_Inquiry* para iniciar el proceso de búsqueda, siempre y cuando se haya activado su generación mediante el comando *HCI_Write_Inquiry_Mode*. Por tanto, eso conduce a pensar que para obtener el valor del parámetro RSSI o bien durante la transferencia de información se lleva a cabo un procedimiento de INQUIRY o dicha lectura se realiza de forma ajena a la conexión que se pretende caracterizar.

También, en [Rashid06] y en [Hipólito09] se mide el retardo de transmisión de forma experimental, mediante la transferencia de ficheros. Concretamente, en [Rashid06] se presenta una aplicación gráfica que permite la comunicación entre ordenadores personales (PC) que disponen de interfaz Bluetooth, utilizando el perfil PAN. Con esta aplicación han realizado algunas medidas de retardo de transmisión durante el envío de ficheros,

utilizando diferentes dispositivos Bluetooth y modificando la distancia entre dispositivos y los obstáculos intermedios. En las conclusiones los autores indican que el retardo crece exponencialmente con el tamaño del fichero a transmitir, y que aumenta con la distancia entre transmisores. Al respecto cabe señalar que esta conclusión parece indicar que el retardo medido no está computando únicamente el consumido en la transmisión Bluetooth, es decir, está enmascarando el efecto de otros factores que no se han tenido en cuenta como, por ejemplo, la propia aplicación.

En [Hipólito09] se presentan los resultados de medidas realizadas utilizando dispositivos Bluetooth comerciales. Las medidas realizadas se llevan a cabo configurando una *piconet* Bluetooth dentro de un entorno de interior utilizando, probablemente (dado que no se especifica), el perfil PAN. El parámetro medido es el retardo de transmisión de un fichero de diferentes tamaños. Las medidas se han repetido modificando la separación entre nodos de la *piconet* e intercalando obstáculos. El control de la medida se ha llevado a cabo utilizando *software* de monitorización (*sniffers*), posiblemente a nivel IP. Las conclusiones del experimento se limitan a obtener la recta de regresión de las medidas realizadas según los parámetros de medida. Igualmente se indica que en presencia de una red 802.11 corporativa las medidas no se ven afectadas de forma significativa, a pesar que la literatura al respecto indica lo contrario. Aunque se realizan medidas con dispositivos reales, el experimento mostrado no es suficientemente realista, ni tampoco útil para obtener un modelo empírico. Además, no se ha llevado a cabo con rigor suficiente para que pueda ser repetible.

Por último, cabe citar el trabajo presentado en [Perrucci09] en el que se describe cómo la elección del paquete apropiado contribuye a reducir el consumo energético. El estudio se ha llevado a cabo experimentalmente con dispositivos reales, concretamente terminales móviles (Nokia N71 y N95). Para ello se ha utilizado la herramienta de Nokia *Energy Profile* [Nokia] y se ha desarrollado una aplicación en lenguaje Python [Python] que hace uso de una extensión en Symbian C++ para configurar el tipo de paquete que se desea emplear. Como resultado de las medidas realizadas los autores proporcionan una tabla en la que se indica para un número de octetos dado (desde 0 hasta 350 octetos), cuál sería el paquete que menos consumo energético requeriría.

3.3. Estudios de rendimiento relacionados con la coexistencia IEEE 802.11-Bluetooth

Desde el año 2001 y hasta el año 2004, en el área de estudio de la coexistencia, cabe mencionar las diversas contribuciones de la profesora Nada Golmie [Golmie01c, Golmie01a, Golmie01b, Golmie02, Golmie03b, Golmie03d, Golmie03a, Golmie03c, Golmie03e, Golmie04]. Concretamente, cabe destacar el trabajo presentado en [Golmie04] como el más reciente del conjunto en el que se aborda esta línea que ha suscitado tanto interés, que además es posterior a la publicación de la versión Bluetooth 1.2 en la que se incluye ya la técnica AFH como mecanismo de coexistencia. En [Golmie04] se presenta un algoritmo de planificación para Bluetooth, que la autora designa como BIAS (*Bluetooth Interference Aware Scheduling*), cuyo objetivo es minimizar el efecto de la interferencia sobre Bluetooth adaptándose a los cambios en las condiciones del entorno. El algoritmo propuesto contempla que los miembros de una *piconet* no experimentan el mismo nivel de interferencia en todos los canales de la banda. Así, Golmie propone modificar la secuencia de salto, distribuyendo los canales entre los distintos dispositivos de forma que se maximice la tasa efectiva de transmisión, al tiempo que se asegure la distribución equitativa entre los usuarios asignando prioridades. La evaluación de rendimiento se realiza mediante simulación, utilizando OPNET [OPNET], para cuatro escenarios: (1) caso básico, sin interferencia e introduciendo 1, 2 y hasta 3 sistemas WLAN para generar interferencia; (2) comportamiento dinámico, que se simula encendiendo y apagando las fuentes de tráfico WLAN unas 30 veces, durante el tiempo de simulación; (3) se evalúa el soporte de QoS considerando diferentes niveles de interferencia para los dispositivos y requisitos de calidad distintos para las conexiones establecidas; (4) a la interferencia WLAN se añade la interferencia introducida por otras *piconets* existentes en el entorno. La autora concluye que el algoritmo propuesto mitiga la pérdida de paquetes, incluso cuando en el peor de los casos se solapan más del 75 % del espectro. Además, el algoritmo BIAS es capaz de adaptarse rápidamente a los cambios en el canal, con un tiempo de transición de 1,5 ms y de 250 μ s para paquetes DH5 y DH1, respectivamente. Por contra, reconoce que no es efectivo para minimizar la interferencia causada por otras *piconets* Bluetooth.

Acerca del impacto sobre el rendimiento Bluetooth de la interferencia provocada por

redes WLAN versa el trabajo presentado en [Karjalainen02], en el que se incluyen los resultados de las medidas realizadas en tres escenarios: (1) conectando los dispositivos al mismo PC; (2) separando los receptores, mediante paredes físicas, y manteniendo juntos los transmisores, de forma que cada uno de los receptores (WLAN y BT) tiene en línea de visión directa a su correspondiente transmisor; (3) separando también los transmisores. A partir de las medidas realizadas, en las que se modifican las distancias entre los distintos elementos, los autores observan que la degradación en la tasa efectiva de transmisión es de un 28 % cuando sólo un dispositivo WLAN está interfiriendo en la transmisión Bluetooth, incrementándose hasta un 45 % si se produce interferencia tanto desde el transmisor como desde el receptor WLAN, siempre y cuando los dispositivos Bluetooth no se alejen más de 5 m. Finalmente, se concluye que con una simple planificación de WLAN y Bluetooth, ambas tecnologías pueden coexistir con unos niveles aceptables de degradación del rendimiento.

En [Jiang03] se recopilan los resultados de simulaciones de transmisión de voz en una *piconet* Bluetooth, realizadas utilizando el simulador OPNET, junto con rutinas de codificación de voz en Matlab. El objetivo es la evaluación de la calidad de voz, empleando la medida *Mean Opinion Score* (MOS), obtenida en presencia de interferencia 802.11b. Los resultados se han obtenido caracterizando las transmisiones Bluetooth mediante parámetros relacionados con la transmisión (modulación, potencia transmitida, etc.) y la codificación de los paquetes de voz, incluyendo protección frente errores. Las fuentes interferentes 802.11b se modelan, básicamente, mediante parámetros de transmisión. El modelo de canal incluye la atenuación, dependiente de un parámetro, debida a la distancia entre transmisores y la adición de ruido gaussiano (canal AWGN). Los resultados expuestos muestran la medida MOS ante variaciones de la distancia entre el receptor Bluetooth y la fuente interferente 802.11b, para distintas modulaciones de 802.11, concluyendo que es necesaria una distancia a la fuente interferente superior a 5–7 m para garantizar una calidad aceptable.

En [Arumugam03] los autores extienden los resultados de [Arumugam02] para estudiar la coexistencia entre Bluetooth y IEEE 802.11g. Partiendo de algunos de los modelos presentados en el anterior artículo, incorporan el modelo de interferencia para IEEE 802.11g y realizan simulaciones con el fin de obtener, para Bluetooth, gráficas de la tasa PER (*Packet Error Rate*) en función de la relación SNR, para distintas relaciones entre portadora e interferente (CIR, *Carrier-Interference Ratio*). Finalmente proponen la utilización de

diversidad espacial y códigos de bloque espacio-temporales (STBC, *Space-Time Block Code*) para mitigar el efecto de la interferencia y el canal.

En [Francia04] se estudia, con un enfoque empírico, el rendimiento de Bluetooth evaluándolo en términos de tasa de paquetes perdidos para varios escenarios resultantes de combinar los modos de operación de IEEE 802.11: modo *ad hoc*, modo infraestructura y ambos combinados con Bluetooth. A partir de las medidas realizadas los autores concluyen que la tasa de paquetes perdidos es mayor cuando IEEE 802.11 funciona en modo *ad hoc* que cuando lo hace en modo infraestructura, correspondiendo el caso peor al escenario en el que coexisten de forma simultánea los modos *ad hoc* e infraestructura con Bluetooth.

El trabajo presentado en [Cordeiro04] también tiene un carácter práctico. Concretamente, los autores proponen una arquitectura, designada con el nombre *BlueStar*, en la que los dispositivos que integran interfaces Bluetooth y 802.11, denominados BWGs (*Bluetooth Wireless Gateways*) permiten a dispositivos Bluetooth pertenecientes a una *piconet* o *scatternet* acceder a una red WLAN prescindiendo del uso de puntos de acceso Bluetooth fijos. Adicionalmente, para mitigar el impacto de la interferencia provocada por las comunicaciones IEEE 802.11, optan por aplicar, de forma combinada, las técnicas AFH y de detección de portadora Bluetooth (BCS, *Bluetooth Carrier Sense*). En relación con la arquitectura propuesta, los autores también proporcionan una cota superior del número de BWG requeridos en una *scatternet* Bluetooth para garantizar acceso ininterrumpido a la red inalámbrica.

En [Bamahdi06] se propone un nuevo esquema adaptativo AFH para mitigar el efecto sobre Bluetooth de la interferencia provocada por las transmisiones de las redes WLAN. Dicha técnica optimiza la separación entre canales de acuerdo con la carga de red y el nivel de ruido. Los autores explican que, para un ancho de banda y una tasa de transmisión dados, reducir la separación entre canales adyacentes permite aumentar el número de canales disponibles, disminuyendo la probabilidad de colisión. Como contrapartida, aumenta también la interferencia cocal. Teniendo presente esta dualidad, los autores apuntan que existe un espaciado óptimo que maximiza la tasa efectiva de transmisión. Esta es precisamente la clave del algoritmo que proponen, cuya eficiencia la prueban, mediante simulación, considerando un canal Rayleigh y un rango amplio de cargas de red. A partir de los resultados concluyen que el espaciado óptimo está comprendido entre 0,5 y 0,75 MHz.

Sin embargo, esta conclusión implicaría también una modificación del estándar que, además, en este caso no se considera acertada por el motivo indicado por los propios autores: la interferencia cocanal.

En [Pang06] los autores también tienen como objetivo mejorar el mecanismo AFH introducido en la especificación Bluetooth 1.2. Partiendo del supuesto de interferencia debida a WLAN 802.11b, proponen dos técnicas: (1) marcar por grupos los canales afectados por un único canal 802.11b; (2) flexibilizar el mecanismo de exclusión de canales, introduciendo un factor probabilístico de selección de canales previamente eliminados. Ambas propuestas se validan mediante simulación.

En [Li07a] se pone de manifiesto que la interferencia cocanal ha llegado a convertirse en un problema debido al creciente despliegue de redes inalámbricas en la banda ISM. Alternativamente a la solución comunmente adoptada por Bluetooth para evitar colisiones con transmisiones WLAN (modificar la secuencia de salto), los autores proponen para Bluetooth el empleo de una técnica de diversidad en frecuencia, denominada DCT (*Dual Channel Transmission*), cuyo objetivo es minimizar la tasa de paquetes perdidos (PER). Esta técnica, que representa también el núcleo de los trabajos presentados por los autores en [Li07b, Li07c] y [Li09], consiste en transmitir simultáneamente el mismo paquete a dos frecuencias distintas, empleando para cada canal la mitad de la potencia que se requeriría para la transmisión en un canal único (SCT, *Single Channel Transmission*). La validación del análisis realizado se lleva a cabo, mediante simulación, empleando el método Montecarlo. Los resultados revelan que, para una relación SNR de 18 dB (equivalente a 15 dB en cada canal de DCT) y un total de 12 *piconets*, con DCT la tasa PER se reduce en un 50%, incrementándose la tasa efectiva de transmisión en un 8 % respecto al logrado con SCT.

El trabajo presentado en [Yin09] también tiene por objetivo minimizar, e incluso cancelar, la interferencia de Bluetooth sobre comunicaciones IEEE 802.11b. Concretamente, los autores proponen un cancelador de interferencia de banda estrecha consistente en el empleo de un filtro adaptativo seguido por un receptor *rake* combinado, para mejorar la eficiencia, con la utilización de un algoritmo LMS modificado. Los resultados de la simulación realizada incluyen gráficas en las que se representa la tasa BER medida para IEEE 802.11b en función de la relación SIR (*Signal-to-Interference Ratio*) y ponen de manifiesto que se consigue una reducción significativa de la tasa BER con el sistema propuesto.

Adicionalmente, con el propósito de optimizar el rendimiento de Bluetooth en presencia de interferencia 802.11, en [Taher09] se propone un algoritmo que complementa el esquema AFH. El algoritmo propuesto, mediante la detección de colisiones y empleando la información de la secuencia de salto, pretende detectar cuándo un punto de acceso IEEE 802.11 está operando y el canal que está ocupando, con el fin de evitar la región espectral en la que se produce el solapamiento. Los autores, mediante la ejecución de simulaciones en Matlab, evalúan la mejora conseguida, comparándola con los resultados obtenidos aplicando exclusivamente el esquema básico AFH y, también, prescindiendo de su uso. El estudio concluye que con el método propuesto se consigue un comportamiento 5 veces mejor que con AFH y de 25 a 45 veces respecto al caso en el que no se utiliza mecanismo de protección alguno.

En [Mathew09] los autores estudian la coexistencia de Bluetooth 1.1 y WLAN 802.11 desde el punto de vista de la transmisión de voz. Partiendo de la premisa de que las transmisiones de voz se ven severamente afectadas por la interferencia de redes WLAN, pretenden utilizar un tipo de paquete denominado SCORT (SCO con retransmisión), propuesto en el año 2001 por el grupo IEEE 802.15-1 dedicado al estudio de la coexistencia entre Bluetooth y WLAN. Los resultados se obtienen utilizando un modelo disponible para Matlab/Simulink de la transmisión Bluetooth afectada por interferencia WLAN. Sin embargo, los autores obvian que la versión Bluetooth 2.0 + EDR, publicada en el año 2004, incluye en su definición paquetes para la transmisión de voz con codificación robusta y mecanismos de retransmisión, que superan la propuesta planteada en este artículo.

El objetivo de [Roy10] es también el estudio del rendimiento de Bluetooth en presencia de redes WLAN, considerando los modelos de canal AWGN y Rayleigh. Adicionalmente, en [Roy10] también se analizan las distintas técnicas existentes para minimizar el impacto de la interferencia: (1) adaptación de la secuencia de salto, (2) control de potencia y (3) combinación de ambas. Por último, a partir de las simulaciones realizadas, los autores concluyen que cuando no hay limitación de potencia máxima de transmisión, pueden emplearse cualquiera de las dos primeras técnicas, de forma que en caso contrario es más adecuado optar por modificar la secuencia de salto. No obstante, los autores resaltan que en caso de decantarse por la primera opción, hay que tener en cuenta que incrementar en demasía la potencia puede provocar una distorsión significativa sobre las transmisiones

IEEE 802.11. Adicionalmente apuntan que, para niveles elevados de interferencia, la alternativa idónea es simultanear la aplicación de ambas técnicas.

En [Conti03] se presenta un modelo analítico, basado en canales de tipo Rayleigh y Rice, para evaluar analíticamente el rendimiento de Bluetooth cuando es interferido por 802.11 y viceversa. El modelo presentado calcula la probabilidad media de error de paquete y la validan, mediante simulación, considerando varios modelos de tráfico de Bluetooth (voz, tráfico de tipo binomial, etc).

En [Ophir04] se analiza la coexistencia entre IEEE 802.11 y Bluetooth desde una perspectiva muy diferente: cuando los transceptores de ambas tecnologías se integran en un mismo dispositivo. El objetivo de los autores es evaluar su rendimiento cuando se emplea el mecanismo colaborativo TDM (*Time Division Multiplexing*), que permite compartir el tiempo y el medio entre las conexiones IEEE 802.11 y Bluetooth. La evaluación llevada a cabo experimentalmente contempla tres escenarios: (1) Rendimiento de cada tecnología operando de forma independiente (con el transceptor de la otra interfaz apagado); (2) Rendimiento de cada tecnología sin introducir técnica de coexistencia; (3) Rendimiento de cada tecnología con ambos transceptores operativos y empleando la técnica TDM. A partir de los resultados, los autores resaltan la importancia de utilizar mecanismos de coexistencia, ya que en caso de prescindir de estos, aunque Bluetooth es capaz de operar con un 80 %-90 % del máximo rendimiento, lo hace a costa de la disfunción de las comunicaciones WLAN. De hecho, como los autores apuntan en un escenario real, esto supone la pérdida de la conexión 802.11, de forma inmediata, tras el establecimiento de una conexión Bluetooth, dándose en algunos casos, también la situación contraria. Por contra, el uso de TDM permite garantizar el funcionamiento simultáneo de ambas tecnologías, aunque el rendimiento cae a un 60 % respecto del máximo logrado de forma independiente. Por último, cabe resaltar que los autores advierten que Bluetooth perjudica las comunicaciones WLAN, más que a la inversa.

Precisamente, el efecto de la interferencia Bluetooth sobre IEEE 802.11b se analiza en [Jo03]. Para ello, se evalúan, en términos de tasa efectiva de transmisión, los diferentes modos de funcionamiento de la capa MAC de IEEE 802.11 frente a la carga de tráfico ofrecida, estimando la tasa efectiva de transmisión no sólo como una función de los parámetros WLAN y de la distancia, sino también en función de otros parámetros, como la densidad de transmisores en un área determinada y parámetros radio de Bluetooth. Los

autores comienzan presentando el modelo de interferencia desarrollado, basado en trabajos anteriores, en función de la probabilidad de colisión en tiempo y frecuencia. Después, a partir de los resultados obtenidos mediante simulación, concluyen que la tasa efectiva de transmisión Bluetooth respecto a la obtenida en condiciones ideales puede degradarse entre un 25 % y un 66 %, dependiendo de que la tasa de transferencia WLAN sea de 2 Mbps o de 11 Mbps.

En [Howitt03] se desarrolla un método analítico para evaluar la influencia de las transmisiones de *piconets* Bluetooth en el rendimiento de IEEE 802.11b. El método propuesto se basa en el cálculo de la probabilidad de colisión, atendiendo a las características espectrales y temporales de ambas tecnologías.

En [Navarro-Garcia05], donde también se estudia el impacto de Bluetooth sobre comunicaciones WLAN, se considera un escenario de evaluación integrado por varios usuarios de WLAN operando en modo DSSS (*Direct-Sequence Spread Spectrum*) y un transceptor Bluetooth. Para este escenario, considerando el modelo de canal de Rayleigh, se obtiene una expresión para la probabilidad de error que es evaluada, para distintas distancias, mediante simulaciones de Montecarlo. Los autores concluyen, a partir de la simulaciones realizadas, que la degradación es inaceptable si la separación entre un receptor WLAN y el transmisor Bluetooth es inferior a los 7 m.

En la misma línea, en [Hsu06] los autores presentan un método de coexistencia entre 802.11 y Bluetooth no cooperativo consistente en optimizar dinámicamente, en función de la tasa PER, la longitud del paquete de la capa MAC IEEE 802.11 a utilizar para minimizar el efecto de la interferencia causada por Bluetooth. Los autores concluyen, a partir de las simulaciones realizadas, que se consigue un incremento significativo en el rendimiento de 802.11, tanto en términos tasa efectiva de transmisión como de retardo de transmisión.

En [Gopalpet07] y [Song07], teniendo en cuenta que las redes WLAN y WPAN (*Wireless Personal Area Network*) proporcionan servicios complementarios en la misma banda de frecuencia y que, junto al aumento de las ventajas derivadas de la combinación de ambos servicios aumenta también la probabilidad de la interferencia mutua, los autores proponen una modificación del esquema AFH, al que denominan IAFH (*Interference-Aware Adaptive Frequency Hopping*). El objetivo es minimizar el impacto de Bluetooth sobre las transmisiones 802.11. El análisis y el modelo analítico de interferencia realizado para el algoritmo

propuesto, en función de la probabilidad de colisión, se valida mediante simulaciones realizadas en Matlab en las que se comprueban sus ventajas frente al esquema especificado en Bluetooth.

En [Cabral09] también se estudia el impacto de la interferencia causada por Bluetooth sobre las transmisiones de datos IEEE 802.11g, en este caso, empíricamente y considerando un entorno doméstico. Para ello, el escenario de test empleado por los autores está integrado por dos PC conectados a un punto de acceso y dos teléfonos móviles dotados con tecnología Bluetooth, de forma que el transmisor y receptor Bluetooth se han colocado, respectivamente, junto a los transmisores y receptores WLAN. En este escenario las medidas se han efectuado, modificando la separación entre los nodos WLAN y, enviando un fichero de 715 MB y otro de 30 MB para generar, respectivamente, el tráfico WLAN y Bluetooth. A partir del análisis de tres parámetros —tiempo de transmisión, tasa media de transmisión y número de paquetes perdidos— los autores concluyen que la interferencia provocada por Bluetooth tiene un mayor impacto sobre el rendimiento de 802.11 conforme aumenta la distancia y la presencia de obstáculos entre los nodos WLAN. Los autores apuntan que puede deberse a que Bluetooth, en presencia de interferencia, emplea paquetes de tipo DM1, lo que perjudica más que si se utilizan paquetes DM3 ó DM5 debido a que cambia más la frecuencia de transmisión, ya que durante el envío de paquetes *multi-slot* no se modifica la frecuencia de transmisión.

En [Zielinski07], siguiendo un enfoque empírico, se ilustra cómo el funcionamiento de Bluetooth degrada el rendimiento de las comunicaciones IEEE 802.11. Para ello, considerando que la red WLAN opera en modo *ad hoc* y en modo infraestructura, los autores realizan medidas utilizando el protocolo ICMP (*Internet Control Message Protocol*) y el programa QCheck [IxChariot] destinadas a evaluar el rendimiento de los protocolos TCP y UDP. A partir de los resultados, los autores concluyen que el modo *ad hoc* es el más sensible a la existencia de interferencia.

En [Issac09] el objetivo es evaluar el rendimiento de transmisión en diversos escenarios de coexistencia entre nodos 802.11 de diferentes estándares, incorporando interferencia Bluetooth. El autor presenta los resultados de simulaciones de redes IEEE 802.11, realizadas con la herramienta OPNET, aunque lo que resulta llamativo es que considera un número de esclavos por maestro Bluetooth incompatible con el estándar.

En [Xie09] los autores proponen un sistema de gestión centralizada para balanceo de carga en redes WLAN 802.11 con múltiples dominios en entornos con interferencia generadas por otras redes WLAN así como por redes WPAN Bluetooth. El objetivo es desarrollar un sistema coordinador que optimice el reparto de tráfico entre dominios WLAN, de forma que se mitigue el efecto de la interferencia generada por redes WLAN y WPAN Bluetooth. La validación del sistema propuesto se realiza mediante simulación, utilizando una herramienta propia. Los autores destacan la mejora sustancial del tráfico cursado gracias al sistema de gestión centralizada.

3.4. Relación entre la probabilidad de error y la relación SNR para las modulaciones Bluetooth

La presente sección tiene por objetivo ofrecer una síntesis de todos los trabajos de investigación que se centran, directa o indirectamente, en la relación entre la probabilidad de error y la relación SNR para las modulaciones empleadas por la tecnología Bluetooth —GFSK, $\pi/4$ DQPSK y 8 DPSK— y que han sido estudiados con el fin de desarrollar, para Bluetooth 2.0 + EDR, el cálculo del retardo de transmisión teniendo en cuenta el nivel de ruido existente.

En el apartado 3.4.1 se comienza repasando los artículos en los que se presentan el diseño e implementación de demoduladores para la tecnología Bluetooth, que son evaluados en términos de parámetros como la tasa BER o la relación SNR. A continuación, en el apartado 3.4.2, se recopilan los artículos en los que se abordan los fundamentos teóricos sobre la obtención de la probabilidad de error a partir de la relación SNR para distintas modulaciones y diferentes modelos de canal, independientemente de la tecnología. Después, en el apartado 3.4.3, se repasan los que, siguiendo un enfoque general, ofrecen los resultados de la aplicación del tema investigado a la tecnología Bluetooth y, en última instancia, en el apartado 3.4.4, se engloban los que se centran específicamente en Bluetooth.

3.4.1. Diseño y fabricación de moduladores y demoduladores

En [Xin02, Xia03] los autores presentan el diseño de un demodulador GFSK para Bluetooth utilizando tecnología CMOS. Los autores destacan la robustez del diseño y las prestaciones en cuanto a consumo y tasa BER. Además de validar el diseño mediante simulación, muestran los resultados experimentales de un prototipo.

En [Sheng03] se presenta el diseño de un receptor completo Bluetooth en tecnología CMOS. Los autores analizan, proporcionando detalles de la implementación, los diferentes elementos que integran el receptor, y aportan datos de rendimiento tales como la sensibilidad para una tasa BER dada, o la figura de ruido.

En [Hyun04] los autores presentan un transceptor para Bluetooth EDR desarrollado en tecnología CMOS de $0,25\mu\text{m}$. Este transceptor permite utilizar los tres tipos de modulación disponibles en el estándar Bluetooth 2.0 + EDR. Los autores describen las soluciones adoptadas para evitar diversos problemas en el diseño y proporcionan resultados experimentales sobre el rendimiento esperado: sensibilidad, rechazo de señal interferente, consumo, etc.

En [Lee06] los autores presentan un demodulador de señal mixta analógico-digital (*mixed-signal*) para GFSK realizado también con tecnología CMOS. El diseño es comparado, mediante simulación, con otros tipos de demoduladores típicos para GFSK, en términos de parámetros de rendimiento como, por ejemplo, la tasa BER lograda para una relación SNR dada.

En [Kao07] se presenta el diseño de un demodulador GFSK en tecnología CMOS, utilizando un DSP (*Digital Signal Processor*) y líneas de retardo con autocalibración. Los autores indican parámetros de rendimiento, como la relación SNR para un límite de tasa BER dado o la máxima desviación de frecuencia detectable, resultando ser todos compatibles con el estándar Bluetooth.

En [Ibrahim07] los autores proponen el diseño de un receptor Bluetooth basado en la descomposición de Laurent. Los autores destacan las ventajas que permite lograr este enfoque, en concreto su baja complejidad, robustez y buen rendimiento. En el artículo

exponen además el desarrollo analítico del receptor y evalúan su rendimiento en diversas circunstancias, mediante simulación, comparándolo con otros métodos.

En [Han08] y [Han09] los autores presentan el desarrollo de un demodulador GFSK en tecnología CMOS de 0,18 μm . El diseño se ha realizado teniendo en cuenta los requisitos de tecnologías WPAN como Bluetooth, por lo que presentan resultados de rendimiento del diseño en términos de SNR mínima para una velocidad de transmisión e índice de modulación dados, según la tasa BER deseada. Adicionalmente, proporcionan información sobre el consumo y la ocupación de área del diseño. Los resultados son comparados con los conseguidos con otros diseños similares.

3.4.2. Estudios teóricos genéricos

En [Stein64] se realiza un análisis unificado de las algunas modulaciones binarias con detección coherente y no coherente. Se trata de un artículo clásico en el que aún se siguen basando desarrollos posteriores sobre comunicaciones digitales en canales gaussianos.

En [Wong92] se estudia la aplicación de las técnicas de corrección de errores no redundantes a la modulación $\pi/4$ DQPSK en canales gaussianos. Los autores realizan un análisis teórico de estas técnicas, concluyendo que aportan mejoras en el rendimiento. Algunos resultados son validados mediante simulación.

En [Ng93] los autores obtienen fórmulas analíticas de la tasa BER para la modulación $\pi/4$ DQPSK en canales de tipo Rayleigh, y comparan su rendimiento con la modulación DQPSK bajo diferentes valores del parámetro BT (producto del ancho de banda por el tiempo).

En [Noneaker94] los autores estudian las modulaciones M-PSK y M-DPSK bajo canales con desvanecimientos, de tipo Rice, y con diversidad, obteniendo expresiones que acotan de forma ajustada la tasa BER para BPSK, 8 DPSK y 16 DPSK.

En [Chow94] se estudia el rendimiento de modulaciones M-DPSK con filtrado en coseno alzado (*raised-cosine*, RC), en canales Rayleigh, incluyendo diversidad en recepción. El objetivo es comparar la modulación para distintos números de símbolos, proporcionando una cota superior en el caso considerado. Los autores, mediante simulación numérica,

concluyen que 4 y 8 DPSK se comportan mejor que B-DPSK y 16 DPSK, por lo que se pueden considerar modulaciones óptimas.

En [Kiasaleh96, Kiasaleh97] se realiza el análisis teórico de la modulación $\pi/4$ DQPSK, introduciendo factores de error como desequilibrios en el mezclador o errores de temporización. Se calculan expresiones que permiten obtener la tasa BER partiendo de la relación SNR en un canal de tipo Rayleigh al introducir los factores de error mencionados. Se concluye comparando, a partir de resultados numéricos, el comportamiento de esta modulación con la modulación DQPSK bajo diversas combinaciones de los efectos estudiados.

En [Shimizu97] los autores presentan un método analítico para el cálculo de la tasa BER para la modulación GFSK considerando el efecto del filtro de postdetección. Los cálculos analíticos se validan mediante simulación numérica, aunque realizan algunas medidas sobre prototipos. Los resultados de este artículo aproximan la tasa BER real de la modulación GFSK ya que sí tienen en cuenta el efecto de la interferencia intersimbólica (ISI, *Inter-Symbol Interference*).

En [Simon98] se presenta un enfoque unificado para el análisis del rendimiento de modulaciones digitales en canales con desvanecimientos. Este artículo destaca especialmente porque constituye un referente debido a que proporciona una formulación uniforme, utilizando las funciones gaussiana y Marcum-Q, para la obtención de la tasa BER correspondiente a los principales tipos de modulaciones con detección coherente, no coherente y diferencial, para los modelos de canal frecuentemente utilizados en comunicaciones digitales.

En [Miller98] se presentan expresiones que calculan la tasa BER para la modulación $\pi/4$ DQPSK. También se analiza el comportamiento de esta modulación en canales de tipo gaussiano y Rayleigh, facilitándose expresiones generales que se particularizan para diversos casos, incluyendo extensiones para recepción con diversidad.

En [Vitetta97b, Vitetta97a, Vitetta98, Vitetta99a, Vitetta99b] los autores estudian la modulación FSK en canales con desvanecimiento rápido de tipo Rayleigh y Rice. Partiendo de modelos lineales variantes con el tiempo para los desvanecimientos, se proporcionan expresiones analíticas de la probabilidad de error de bit que son validadas mediante simulación.

En [Chandra08] los autores desarrollan expresiones que permiten determinar la tasa BER, a partir de la relación SNR por bit, para la modulación $\pi/4$ DQPSK considerando recepción simple y multi-canal para un canal genérico con desvanecimientos en presencia de ruido de tipo gaussiano. Con este fin, siguiendo un enfoque totalmente teórico, utilizan los modelos de Rayleigh, Rice, Nakagami- m y Hoit. Además, para mejorar la calidad del enlace tienen en cuenta la diversidad espacial de la antena del receptor, considerando antenas múltiples, independientes e idénticamente distribuidas, de forma que las réplicas de señal no correladas se combinan antes de la etapa de demodulación. Para ello, el combinador de predicción lineal propuesto utiliza el algoritmo MRC (*Maximum Ratio Combining*). Los autores obtienen las funciones de error a partir de la función de densidad de probabilidad como sumas de series, tanto finitas como infinitas, que resuelven mediante integración numérica eligiendo el número de términos necesarios para lograr una estimación fiable de la tasa BER llegando a un compromiso entre precisión y complejidad.

En [Falujah08] se presenta un estudio de las mismas características: se determina la probabilidad de error de bit (BEP, *Bit Error Probability*) media de la modulación DQPSK con postdetección y recepción con diversidad EGC (*Equal Gain Combining*) para canales con desvanecimientos arbitrariamente correlados e independientes. Empleando funciones de Legendre, se obtiene el promedio de la probabilidad BEP para los modelos de Rayleigh, Rice y Nakagami- m , al igual que en [Chandra08], pero proponiendo expresiones en términos de sumas de series finitas, que se validan mediante simulación numérica.

En [Kostic08] se expone el cálculo de la probabilidad de error de símbolo, SEP (*Symbol Error Probability*), para una modulación M-DDPSK (*M-ary Doubly-Differential Phase-Shift Keying*), comparando el rendimiento de una 8 DDPSK con una 8 DPSK en el caso de un canal de tipo Nakagami. El autor concluye que, para valores elevados de la frecuencia portadora y tasas de símbolos bajas, la detección diferencial doble mejora considerablemente el rendimiento de la diferencial básica.

En [Sun09] los autores proporcionan una aproximación a la probabilidad de error de bit de la modulación DQPSK en canales con desvanecimientos. La aportación de este artículo reside en que facilita un método alternativo de cálculo de esta probabilidad de error de bit para canales con desvanecimientos, cuyo cálculo proporciona resultados numéricos mejores que otras aproximaciones propuestas con anterioridad.

3.4.3. Estudios aplicados a Bluetooth

En [García-Armada07], tras realizar una revisión de los modelos de canal a considerar en función de las condiciones de propagación, se recopilan los puntos claves a tener en cuenta para analizar el rendimiento de una red WPAN utilizando herramientas de simulación. Para la estimación de la probabilidad de error se ha empleado el método de Montecarlo. El estudio se complementa con la evaluación de rendimiento empleando cadenas de Markov para dos ejemplos: una *piconet* Bluetooth y una red WPAN basada en OFDM (*Orthogonal Frequency-Division Multiplexing*). El proceso de simulación se realiza combinando simulaciones a dos niveles: nivel de transmisión (*waveform-level*) y nivel del sistema (*system-level*). Del primer nivel se extrae un modelo de canal discreto basado en HMM (*Hidden Markov Model*), que se aplica como componente a la simulación del segundo nivel. En el nivel de transmisión se utilizan parámetros típicos como la relación SNR, modulación empleada, distribución física del transmisor y receptor, fuentes de interferencia, etc. En este nivel se modela la cadena de transmisión como bloques que procesan la información antes de enviarlos por el canal. A partir de esta información, realizan simulaciones (aunque no se indica la herramienta *software* empleada) para obtener el modelo HMM. En el segundo nivel (nivel de sistema) se da a entender que se modelarían las capas superiores de la tecnología que se desearan simular, incorporando el modelo de canal obtenido. Sin embargo, no se dan detalles ni resultados de la simulación de este segundo nivel. De hecho, los resultados incluidos se limitan, para OFDM, a una representación de la tasa BER frente a la relación SNR, y para el modelo de canal discreto Bluetooth, a los resultados numéricos del modelo HMM.

En [Liu03] se presenta un estudio analítico de la modulación GFSK con detección no coherente para sistemas que utilicen salto en frecuencia como técnica de espectro ensanchado, como es el caso de Bluetooth. El análisis se realiza para canales de tipo Rayleigh, obteniendo ecuaciones que expresan la tasa BER en función de la relación SNR, considerando como parámetros el índice de modulación, entre otros. Además, se propone un método para mejorar las prestaciones en sistemas de salto en frecuencia.

En [Tibenderana04] los autores presentan el diseño teórico de un receptor para GFSK con corrección del desplazamiento de portadora. Este diseño presenta un método para simplificar la implementación de un receptor multisímbolo que mejora las prestaciones de

tasa BER, utilizando un algoritmo para compensar el desplazamiento de portadora. Finaliza mostrando algunos resultados para un posible receptor Bluetooth.

En [Chang06], partiendo de la base de que los transceptores GFSK se diseñan normalmente en el dominio analógico y con el fin de lograr un consumo reducido y minimizar su coste, los autores consideran que las tecnologías de demodulación digitales orientadas a mejorar su rendimiento minimizando la tasa BER representan un objetivo de estudio interesante, dado, además, que GFSK es una modulación cuyo uso está ampliamente extendido para la transmisión de datos a baja velocidad en estándares como Bluetooth. En esta línea se propone un transceptor digital para GFSK en el que la novedad aportada es el método de estimación de la frecuencia de portadora. En particular, se propone un sistema de sincronización de portadora simple basado en cancelación de la señal de información, asistida por la etapa de decisión. De esta forma se estima el error de frecuencia de la portadora a partir del sumatorio de las diferencias entre la señal demodulada antes y después de la decisión. Este error se emplea para generar la estimación de portadora en un correlador en cuadratura. Además, mediante simulación, se evalúa el rendimiento del algoritmo, considerando que el canal es de tipo AWGN y suponiendo unas condiciones determinadas, y se concluye que la tasa BER simulada se aproxima al valor teórico ideal. Este es también el objetivo del trabajo presentado, por el mismo autor, en [Chang09]. Concretamente, en este artículo se describe un mecanismo de sincronización de portadora similar al propuesto anteriormente en [Chang06], aplicando en este caso los métodos de mínimos cuadrados (LS, *Least-Squares*) y de máxima verosimilitud (ML, *Maximum-Likelihood*). También, mediante simulación, se evalúa este método comparándolo con el basado en DFT (*Discret Fourier Transform*) y se concluye que su rendimiento es superior al no verse afectado por fuentes de información con componente continua no nula. El análisis parte de la definición matemática de la modulación GFSK y del modelo del receptor generalmente empleado para su detección (correlador en cuadratura). Utilizando el modelo de canal gaussiano, se analiza el comportamiento del demodulador, comparando el método de recuperación de portadora basado en DFT con el que proponen, basado en LS/ML. Tras realizar simplificaciones y suposiciones en el análisis se concluye que la varianza del error para el método propuesto es menor. Finalmente, mediante simulación (aunque no se indica qué herramienta se utiliza), se comparan los distintos métodos expuestos según la tasa BER obtenida para diferentes relaciones SNR. Pese a que la conclusión es que, en determinadas condiciones, la recuperación de portadora según

el método LS/ML puede ser mejor, no se especifican los problemas que pueden aparecer en una implementación real, ni el grado de complejidad computacional requerido.

3.4.4. Estudios específicos de Bluetooth

En [Valenti02b] se analiza el rendimiento de las transmisiones Bluetooth en canales gaussiano y de tipo Rayleigh. En este artículo se obtienen expresiones que permiten obtener la tasa de transmisión alcanzada con cada tipo de paquete Bluetooth para una determinada relación SNR. Para ello se obtiene la tasa PER a partir de la tasa BER correspondiente a la modulación GFSK. Finalmente, se propone un esquema de selección de paquete en función de la relación SNR para canales de tipo gaussiano y Rayleigh. Como errores en el desarrollo analítico, se puede destacar el cálculo erróneo de la probabilidad de error en el código de acceso y un fallo en la formulación de la expresión que determina la probabilidad de error para GFSK.

En [Skudlarek02] el autor presenta un sistema de simulación de RF diseñado para facilitar el desarrollo de un transceptor Bluetooth comercial. El objetivo del simulador es el modelado del camino de transmisión/recepción de forma que sea posible la evaluación y validación de las mejoras que se incorporarán al diseño final. El sistema de simulación se basa en Matlab e integra el comportamiento de circuitos analógicos simulados con *Spice* [Spice].

En [Yoon04] los autores proponen un esquema de selección de paquetes Bluetooth según la estimación de la relación SNR del canal. Los autores proponen nuevas modulaciones para Bluetooth 1.1, no relacionadas con las utilizadas en realidad en Bluetooth 2.0 + EDR. Adicionalmente, proponen algoritmos para la estimación del canal, aunque no valoran la realización práctica por parte de implementaciones reales del nivel LM.

En [Roh06] el autor estudia el rendimiento de transmisiones Bluetooth en canales de tipo Nakagami. El trabajo se limita a la simulación (aunque no indica la herramienta *software* empleada) de transmisiones Bluetooth aplicando el modelo de canal para obtener gráficas de la tasa BER en función de la relación SNR, y el rendimiento de transmisión para diferentes condiciones de interferencia y desvanecimiento.

En [Hua08], repitiendo el desarrollo de [Valenti02b], el autor introduce el modo EDR con las modulaciones asociadas. De igual forma que [Valenti02b], para canal gaussiano, también propone una estrategia de selección de paquetes en función de la relación SNR. Finalmente propone llevar a cabo este mecanismo de selección de paquetes utilizando el modo de retransmisiones del protocolo L2CAP. En cualquiera de los casos no se indica el mecanismo mediante el que el nivel BB selecciona el paquete deseado.

En [Mikulka08] los autores modelan las etapas de transmisión y recepción de un transceptor Bluetooth, incluyendo las modulaciones empleadas y considerando el canal de tipo gaussiano. Para el modelo propuesto, utilizando Matlab, realizan una simulación a fin de determinar la tasa BER correspondiente a diferentes valores de la relación SNR. Como resultados, se incluyen gráficas en las que se comparan las tasas BER teóricas con las obtenidas mediante simulación.

En [Moon07] se realiza un análisis de la transmisión de datos del protocolo de aplicación WAP sobre la pila de protocolos Bluetooth. En este análisis se introduce el efecto de la fragmentación de las PDUs de L2CAP para su transmisión en paquetes BB de diferente tamaño, aunque la explicación de este proceso no es clara ni está justificada según el estándar. Para la obtención de resultados, realiza simulaciones con paquetes DM, incorporando un modelo de canal gaussiano. De igual forma, no parece justificar correctamente el modelo de pérdidas y los resultados de retardo obtenidos no parecen realistas.

3.5. Resumen y conclusiones

La tabla 3.1 esquematiza la clasificación de los artículos que se centran en el estudio del rendimiento de Bluetooth, que han sido analizados a lo largo de las secciones 3.1, 3.2 y 3.3, indicando la metodología seguida en el estudio presentado en cada caso. Además, en la tabla 3.2, para aquellos en los que se lleva a cabo simulación, y los autores lo especifican, se indica cuál ha sido la herramienta utilizada. Al respecto cabe resaltar, en lo que concierne al estudio de los esquemas de transmisión TDD, que los autores de [Misic06] han empleado además del simulador de eventos discretos NS-2, el simulador de redes de Petri, Artifex.

Por su parte, la tabla 3.3 corresponde a una taxonomía de todos los trabajos de investigación, presentados en la sección 3.4, que abordan la relación entre la probabilidad de error y la relación SNR para las modulaciones empleadas por la tecnología Bluetooth —GFSK, $\pi/4$ DQPSK y 8 DPSK—. También se han incluido los artículos que, aunque tratan sobre otras modulaciones, constituyen un referente.

Por último, cabe destacar:

- En cuanto al tema objeto de estudio: Los estudios de rendimiento se realizan atendiendo a factores de muy diversa índole:
 - Esquemas de transmisión TDD.
 - Optimización de funciones, normalmente, pertenecientes al nivel BB, en función de las condiciones del canal.
 - Análisis del impacto de la interferencia, tanto de redes 802.11 sobre el rendimiento de Bluetooth, como a la inversa.
- En relación con la metodología: En los casos en los que se desarrolla un estudio analítico, cuando se acompaña de una validación, ésta siempre es realizada mediante simulación. Por contra, los trabajos que se abordan con un enfoque empírico no están respaldados por un estudio teórico previo. En ellos, tampoco suele derivarse, como resultado de las medidas realizadas, un modelo matemático práctico de comportamiento de Bluetooth.

En definitiva, se echan en falta trabajos de investigación en los que converjan el análisis y modelado de aspectos del protocolo teniendo en cuenta la especificación del estándar y la validación empírica realizada con las implementaciones de dispositivos reales. La conjunción de ambos aspectos (modelo analítico y caracterización empírica) permitiría detectar problemas de rendimiento derivados de decisiones de implementación sobre aspectos que no estén claramente especificados en el estándar.

Tabla 3.1.: Taxonomía de los artículos sobre el rendimiento de Bluetooth

Tema de investigación	Modelado analítico	Modelado analítico con simulación	Estudio empírico
<i>Esquema de transmisión TDD</i>		[Mistic02b]–[Mistic05b]; [Kalvala06, Lee07, Roy07, Zussman07, Reddy08, Zhihong09, Hsu10, Yang10]	
<i>Esquema ARQ</i>	[Zanella09a, Zanella09b]		[Chen04a]
<i>Esquema adaptativo de selección de paquetes</i>		[Chen04b, Chen06, El-Bendary09b, Feng09, Tan09, Xu07]	
<i>Nuevos formatos de paquetes</i>		[Chen06, Xu07, El-Bendary09a]	
<i>Esquemas de codificación</i>		TC: [Valenti02a, Moon08, Kahveci10] IFEC: [Chen04c, Chen06] Otros: [El-Bendary09a]	
<i>Rendimiento TCP/IP</i>		[Chui02, Chen04b]	
<i>Estudios de rendimiento genérico</i>			[Rashid06, Warriach08, Hipólito09, Perrucci09]
<i>Interferencia WLAN sobre BT</i>	Evaluación: [Conti03]	AFH y mejoras de AFH: [Golmie03d, Bamahdi06, Pang06, Taher09] Otras técnicas: [Golmie01a, Golmie01b, Arumugam02, Arumugam03, Golmie02, Golmie03b, Golmie03a, Golmie03c, Golmie04, Mathew09][Li07a]–[Li09] Evaluación: [Golmie01c, Golmie03e, Jiang03, Roy10]	[Karjalainen02, Cordeiro04, Francia04, Ophir04]
<i>Interferencia BT sobre WLAN</i>	Evaluación: [Conti03, Howitt03]	AFH y mejoras de AFH: [Gopalpet07, Song07] Otras técnicas: [Golmie03a, Golmie03c, Xie09, Yin09] Evaluación: [Jo03, Golmie03e, Navarro-García05, Hsu06, Issac09]	[Ophir04, Zielinski07, Cabral09]

Tabla 3.2.: Herramientas de simulación empleadas en los distintos estudios

Tema de investigación	Matlab	NS-2	OPNET	Otros simuladores
<i>Esquema de transmisión TDD</i>		[Mistic02b]–[Mistic05b]; [Hsu10, Reddy08]	[Roy07, Zussman07]	
<i>Esquema adaptativo de selección de paquetes</i>	[Feng09] (Matlab/Simulink)	[Chen04b]		[Chen06]
<i>Nuevos formatos de paquetes</i>				[El-Bendary09a]
<i>Esquemas de codificación</i>	[El-Bendary09a]			[Valentio2a] [El-Bendary09a]
<i>Rendimiento TCP/IP</i>		[Chen04b]		
<i>Interferencia WLAN sobre BT</i>	[Taher09] [Jiang03, Mathew09] (Matlab/Simulink)		[Golmie01c]–[Golmie04]; [Jiang03, Pang06]	[Lio7a]–[Lio9]
<i>Interferencia BT sobre WLAN</i>	[Gopalpet07, Song07]		[Issac09] [Golmie03a, Golmie03c] [Golmie03e]	[Navarro-Garcia05, Xie09]

Tabla 3.3.: Taxonomía de los artículos que versan sobre la relación entre la tasa BER y la relación SNR, para distintas modulaciones

Temas de investigación	GFSK	$\pi/4$ DQPSK	8 DPSK	Otras modulaciones
<i>Diseño y fabricación</i>	[Xin02, Xia03, Sheng03, Lee06, Ibrahim07, Kao07, Han08, Han09]	[Hyun04]		
<i>Estudios teóricos genéricos</i>	[Shimizu97]	[Wong92, Ng93, Kiasaleh96, Kiasaleh97, Miller98, Chandra08, Falujah08, Sun09]	[Chow94, Noneaker94, Kostic08]	[Stein64, Simon98] [Vitetta97b]–[Vitetta99b]
<i>Aplicados a BT</i>	[Liu03, Tibenderana04, Chang06, Chang09]			
<i>Específicos de BT</i>	[Valenti02b, Skudlarek02, Roh06, Moon07]		[Hua08, Mikulka08]	[Yoon04]

CAPÍTULO 4

Cálculo del retardo de transmisión para el perfil SPP incluyendo el modo EDR

4.1. Introducción

El objetivo de este capítulo es calcular el retardo de transmisión para el perfil SPP en Bluetooth 2.0 + EDR. Para ello, se va a reformular el modelo presentado en [Moron08], que se resume en el apartado 4.2, extendiéndolo para incluir el modo EDR y considerando que:

- L2CAP opera en el modo básico.
- El tiempo de almacenamiento en los *buffers* es despreciable.
- La confirmación de cada paquete Bluetooth se realiza en el siguiente *slot* y con un paquete de un *slot*.

La estimación teórica del retardo de transmisión se va a llevar a cabo, en primer lugar, en el apartado 4.3, asumiendo condiciones ideales, es decir, considerando que no se producen retransmisiones a nivel BB, para a continuación (en el apartado 4.4) completar el modelo considerando el efecto de las posibles pérdidas en las transmisiones.

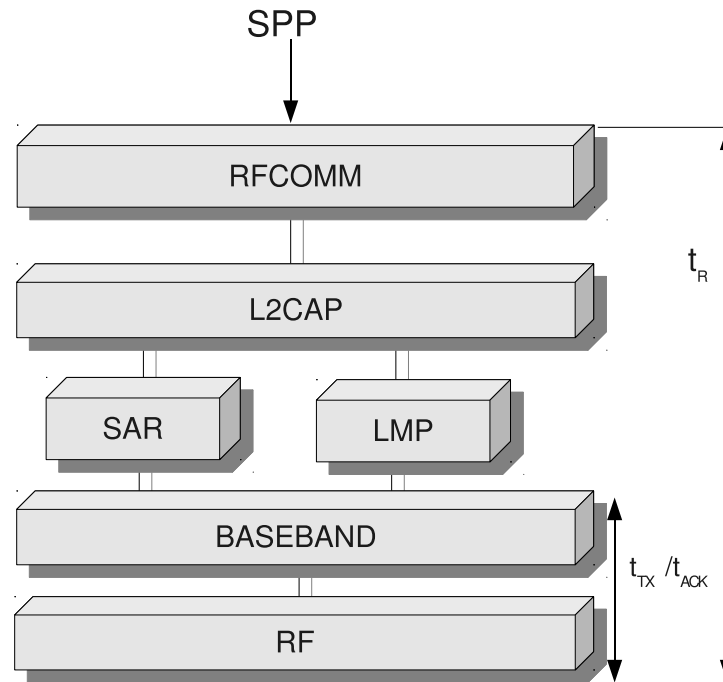


Figura 4.1.: Arquitectura de protocolos Bluetooth (adaptado de: [Moron08])

La figura 4.1 muestra, sobre la arquitectura de protocolos Bluetooth, el retardo de transmisión que se va a calcular para el perfil SPP, t_R , a partir de las componentes t_{TX} y t_{ACK} , computando la sobrecarga añadida por cada uno de los protocolos que intervienen en la transmisión al utilizar dicho perfil. Las componentes t_{TX} y t_{ACK} se definen teniendo en cuenta que los datos del nivel superior a transmitir pueden exceder la máxima capacidad (L_5) de un paquete de 5 slots. Así, estas componentes representan, respectivamente, el tiempo empleado en transmitir cada uno de los paquetes del nivel BB necesarios para enviar N octetos de datos, sin tener en cuenta la confirmación del extremo remoto (t_{TX}) y computándola (t_{ACK}).

Es conveniente aclarar que, para el cálculo de las ecuaciones formuladas a lo largo de los siguientes apartados, se ha utilizado el lenguaje *Python* [Python] y las bibliotecas *NumPy* [NumPy] (cálculo numérico con *arrays*) y *SciPy* [SciPy] (funciones matemáticas especializadas). Estas bibliotecas *open-source* forman parte del conjunto de bibliotecas estándar del lenguaje multiplataforma *Python* y son ampliamente utilizadas para cálculo científico por su rendimiento y calidad de los algoritmos utilizados. Por último, cabe indicar que para representar gráficamente la información computada se ha empleado la herramienta *Gnuplot* [Gnuplot], con licencia GPL.

4.2. Modelo para el perfil SPP en Bluetooth 1.1 en condiciones ideales (sin retransmisiones)

El retardo medio de transmisión de N octetos de usuario utilizando el perfil SPP, en el sentido maestro-esclavo, $\overline{t_{R_M}}(N)$, teniendo en cuenta el efecto de la fragmentación y la sobrecarga introducida por los protocolos RFCOMM y L2CAP, viene dado por la siguiente ecuación, definida para Bluetooth 1.1 [Moron08]:

$$\overline{t_{R_M}}(N) = \frac{T_{poll}}{2} + n_{nff}(N) \cdot t_{ACK}(M(L_R)) + t_{TX}(M(L_{ff}(N))) \quad (4.1)$$

donde el *offset* de valor $T_{poll}/2$, se corresponde con el tiempo medio transcurrido desde que el *Host* solicita el envío de datos hasta que el nivel BB del maestro direcciona al esclavo transmitiéndole esos datos. Nótese que los valores representados son valores medios, y que $T_{poll}/2$ es precisamente la media de una variable aleatoria que se asume que sigue una distribución uniforme entre 0 y T_{poll} .

En la ecuación 4.1, $M(x)$ representa el resultado de sumar a x octetos de usuario la sobrecarga añadida por el nivel RFCOMM y la cabecera de L2CAP, calculado como:

$$M(x) = x + O_R(x) + H_L \quad (4.2)$$

siendo:

- $O_R(x)$: Número de octetos correspondientes a la sobrecarga, es decir, a la información de control del protocolo RFCOMM, que será 4 ó 5 octetos dependiendo de que la extensión del campo de longitud sea 1 ó 2 octetos, lo cual, a su vez, depende del tamaño en octetos, x , de la trama:

$$O_R(x) = \begin{cases} 4 & 0 < x \leq 127 \\ 5 & x > 127 \end{cases} \quad (4.3)$$

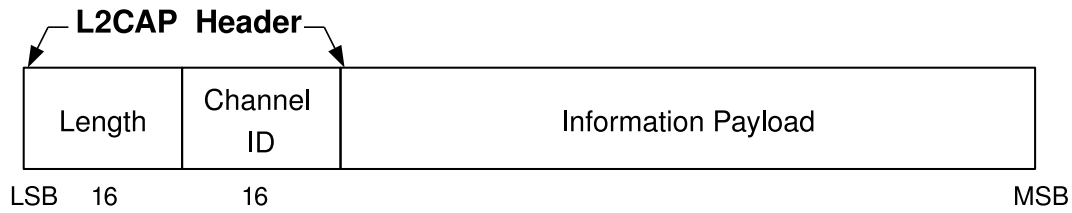


Figura 4.2.: Trama de Información en el modo básico de L2CAP (*B-Frame*) (fuente: [Bluetooth SIG07, vol. 3, parte A]).

- H_L : Considerando que L2CAP opera en el modo básico, son 4 octetos correspondientes al tamaño de la cabecera (*header*) de la PDU de L2CAP, equivalente a la PDU de L2CAP en Bluetooth 1.1 y llamada *B-Frame* (*Basic Information Frame*) en Bluetooth 1.2 y versiones posteriores (véase la figura 4.2).
- L_R : Longitud a la que RFCOMM ha de fragmentar los datos de usuario, garantizando, por un lado, que al sumar la máxima sobrecarga — $\max(O_R(x))$ — que puede introducir RFCOMM, 5 octetos, no se excede la MTU de L2CAP, M_R , y, por otro, que se respeta el valor máximo fijado por el parámetro $N1$.

Así, RFCOMM genera fragmentos de tamaño (véase la figura 4.3):

$$L_R = \min(N1, M_R - 5) \quad (4.4)$$

- $n_{nff}(x) = \left\lceil \frac{x}{L_R} \right\rceil - 1$: Número de fragmentos RFCOMM no finales (*non-final fragments*), teniendo en cuenta que los x octetos de datos se dividen en fragmentos de L_R octetos.
- $L_{ff}(x) = ((x - 1) \bmod L_R) + 1$: Número de octetos correspondientes al último fragmento (*final fragment*). Nótese que, si los datos caben en un único fragmento ($x \leq L_R$), se tendrá que $n_{nff} = 0$ y $L_{ff} = x$.

Para contemplar el hecho de que el número de octetos de la PDU de L2CAP que encapsula la trama de RFCOMM pueda exceder la máxima capacidad de transporte de un paquete DH5 (o, en su caso, DM5), es necesario definir dos componentes del retardo, $t_{ACK}(x)$ y $t_{TX}(x)$, relacionadas tal y como se refleja en la figura 4.4:

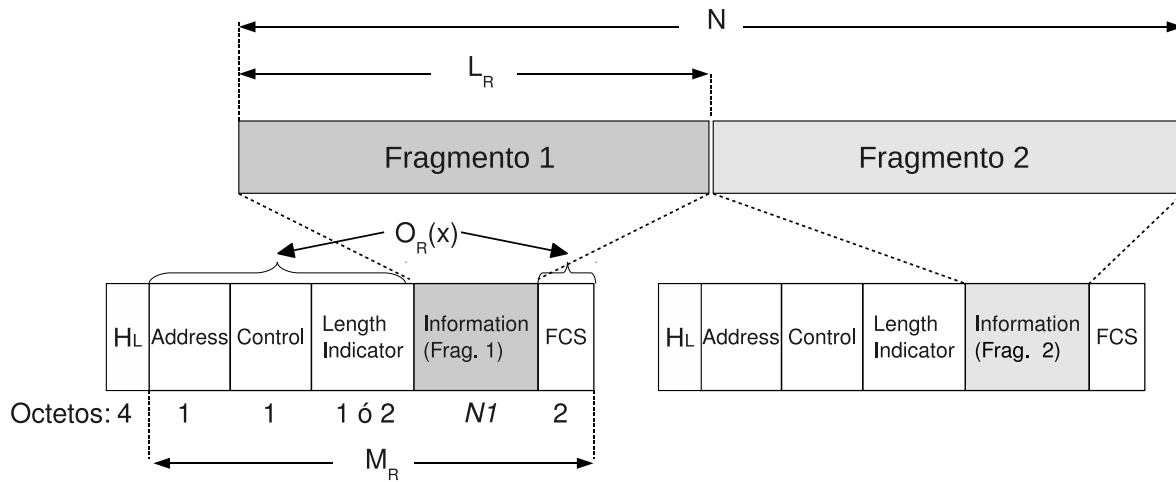
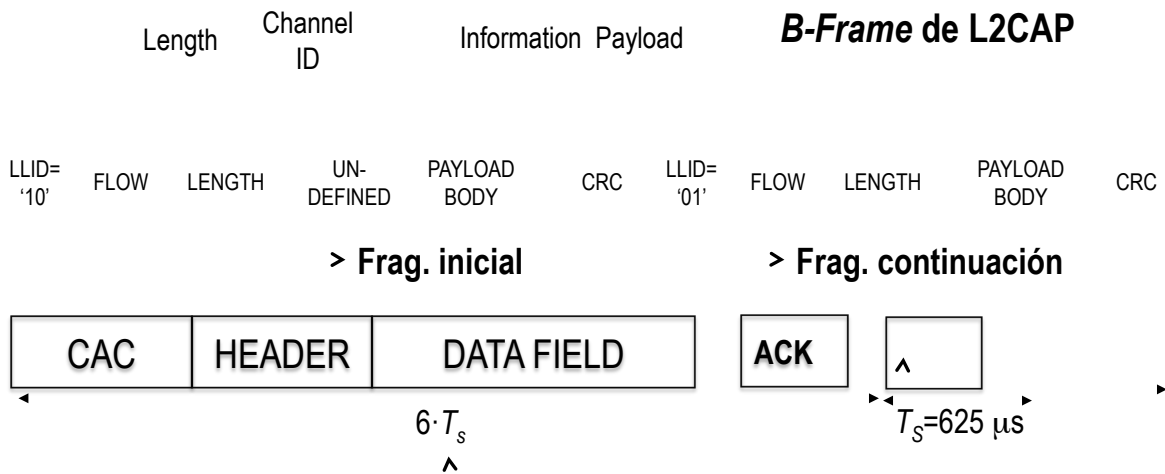


Figura 4.3.: Fragmentación realizada en el nivel RFCOMM cuando el tamaño de los datos de usuario excede el valor de L_R , dado por la ecuación 4.4 (fuente: [Moron08]).



$$t_{TX}(N) = t_{ACK}(L_5) + t_{TX}(N \bmod L_5)$$

Figura 4.4.: Transmisión de una trama de L2CAP cuando se excede la capacidad de un paquete de 5 slots.

- $t_{ACK}(x)$: Representa el tiempo empleado en transmitir y confirmar cada uno de los paquetes del nivel BB necesarios para enviar x octetos de datos. Se calcula mediante

la siguiente expresión recursiva:

$$t_{ACK}(x) = \begin{cases} 0 & x = 0 \\ 2 \cdot T_S & 0 < x \leq L_1 \\ 4 \cdot T_S & L_1 < x \leq L_3 \\ 6 \cdot T_S & L_3 < x \leq L_5 \\ 6 \cdot T_S \cdot \left\lfloor \frac{x}{L_5} \right\rfloor + & x > L_5 \\ + t_{ACK} \cdot (x \bmod L_5) & \end{cases} \quad (4.5)$$

donde el operador $\lfloor Y \rfloor$ redondea el número Y al mayor entero inferior a Y , y el operador $x \bmod y$ obtiene el resto de la división entera de los operandos x e y .

En la fórmula anterior, T_S denota la duración de un *slot* Bluetooth ($625 \mu s$), mientras que L_1 , L_3 y L_5 se corresponden con el máximo número de octetos que pueden transportarse en el campo de datos de un paquete del nivel BB de 1, 3 ó 5 *slots*, que serán, respectivamente, 27, 183 y 339 octetos para los paquetes DH1, DH3 y DH5, y 17, 121 y 224 octetos para los paquetes DM1, DM3 y DM5.

La ecuación 4.5 computa el *slot* de confirmación de la transmisión de cada paquete que transporta un fragmento intermedio, teniendo en cuenta que el siguiente fragmento no va a ser transmitido hasta no haber recibido la confirmación del actual, expresando por tanto que, por cada fragmento intermedio, existe un retardo fijo de 2, 4 ó 6 *slots* dependiendo de que el fragmento en cuestión se transmita en un paquete de 1, 3 ó 5 *slots*.

- $t_{TX}(x)$: Representa el tiempo empleado en la transmisión de x octetos de datos sin considerar el tiempo de confirmación del último (o único) paquete del nivel BB en el que se transporta el último (o único) fragmento del mensaje del nivel superior L2CAP, con independencia de que el esclavo deba enviar un paquete de confirmación en el *slot* siguiente.

La siguiente ecuación realiza el cálculo de este retardo en función del número de bits que se transmiten:

$$t_{TX}(x) = \begin{cases} 0 & x = 0 \\ n_b(x) \cdot t_b & 0 < x \leq L_5 \\ t_{ACK}(L_5) \cdot \left\lfloor \frac{x}{L_5} \right\rfloor + t_{TX}(x \bmod L_5) & x > L_5 \end{cases} \quad (4.6)$$

donde:

- t_b es el tiempo de transmisión de un bit, $1 \mu s$ (teniendo en cuenta que Bluetooth 1.1 transmite a una velocidad de 1 Mbit/s).
- $n_b(x)$ representa el número de bits transmitidos en el último (o único) fragmento y puede expresarse, a su vez, como se indica en la siguiente ecuación:

$$n_b(x) = n_d(x) + O_{BB} \quad (4.7)$$

donde:

- O_{BB} es el tamaño de la sobrecarga (*overhead*) del paquete Bluetooth, definido como la suma de los bits de la cabecera del paquete (54 bits) y del código de acceso (72 bits), lo que supone un total de 126 bits.
- $n_d(x)$ representa el número de bits del campo de datos (*Data Field*), es decir, de la cabecera y cuerpo del *payload*, los 2 octetos de CRC y, en el caso de los paquetes DM, los bits de redundancia añadidos por la codificación FEC (véase la figura 2.4a):

$$n_d(x) = \begin{cases} (x + H_{df}(x) + O_{CRC}) \cdot 8 & \text{paquetes DH} \\ \left\lceil \frac{(x + H_{df}(x) + O_{CRC}) \cdot 8}{10} \right\rceil \cdot 15 & \text{paquetes DM} \end{cases} \quad (4.8)$$

donde el operador $\lceil Y \rceil$ redondea el número Y al menor entero superior.

En la fórmula anterior, O_{CRC} (2 octetos) contabiliza la sobrecarga asociada al campo CRC, mientras que $H_{df}(x)$ representa el número de octetos correspondientes a la cabecera del campo de datos (*Data Field*), de acuerdo con la expresión:

$$H_{df}(x) = \begin{cases} 1 & x \leq L_1 \\ 2 & L_1 < x \leq L_5 \end{cases} \quad (4.9)$$

La ecuación 4.8 tiene en cuenta que para paquetes DM, en los que se protege la información de usuario y los dos octetos de CRC, con codificación FEC $2/3$, por cada 10 bits de información se agregan 5 bits de redundancia, de forma que si el número de bits no es múltiplo de 10 hay que añadir los bits de relleno necesarios detrás del CRC.

En el sentido esclavo-maestro, el retardo de transmisión, $\overline{t_{R_S}}(N)$, se determina a partir de la ecuación 4.1 sumando el *slot*, (T_S) en el que el esclavo (*slave*) es direccionado por el maestro:

$$\overline{t_{R_S}}(N) = T_S + \overline{t_{R_M}}(N) \quad (4.10)$$

4.3. Adaptación del modelo a Bluetooth 2.0 + EDR en condiciones ideales

Dado que los protocolos de los niveles L2CAP y RFCOMM no se ven modificados sustancialmente en la especificación Bluetooth 2.0 + EDR, la adaptación de la expresión del retardo de transmisión para el perfil SPP, $t_R(x)$, requiere reformular únicamente las componentes $t_{ACK}(x)$ y $t_{TX}(x)$, términos que tienen en cuenta la fragmentación realizada a nivel BB de un mensaje del nivel superior. Para ello, es necesario incluir en la definición de dichas componentes los paquetes específicos de los modos EDR, tal y como se describe a continuación:

- $t_{ACK}(x)$: Representa, de nuevo, el tiempo empleado en transmitir y confirmar cada uno de los paquetes del nivel BB necesarios para enviar x octetos de datos. Para Bluetooth 2.0 + EDR se calcula mediante la siguiente expresión recursiva:

$$t_{ACK}(x) = \begin{cases} 0 & x = 0 \\ 2 \cdot T_S & 0 < x \leq L_1 \\ 4 \cdot T_S & L_1 < x \leq L_3 \\ 6 \cdot T_S & L_3 < x \leq L_5 \\ t_{ACK} \cdot (L_5) \cdot \left\lfloor \frac{x}{L_5} \right\rfloor + & x > L_5 \\ + t_{ACK} \cdot (x \bmod L_5) & \end{cases} \quad (4.11)$$

Como en la formulación de Bluetooth 1.1, T_S denota la duración de un *slot* Bluetooth ($625 \mu s$), mientras que L_1 , L_3 y L_5 corresponden a las longitudes máximas del *payload* de un paquete del nivel BB de 1, 3 ó 5 *slots*, que serán, respectivamente, 17, 121 y 224 octetos para los paquetes DM1, DM3 y DM5, al igual que en Bluetooth 1.1, y 83, 552 y 1021 octetos para los paquetes de máxima velocidad (3 Mbps) 3-DH1, 3-DH3 y 3-DH5.

- $t_{TX}(x)$: Representa, asimismo, el tiempo empleado en la transmisión de x octetos de datos sin considerar el tiempo de confirmación del último (o único) paquete del nivel BB en el que se transporta el último (o único) fragmento del mensaje del nivel superior L2CAP, con independencia de que el esclavo deba enviar un paquete de confirmación en el *slot* siguiente. Las ecuaciones 4.12 y 4.13 realizan el cálculo de este retardo en función del número de símbolos que se transmiten, para paquetes DM, $t_{TX}^{DM}(x)$, y paquetes DH, $t_{TX}^{DH}(x)$, respectivamente:

$$t_{TX}^{DM}(x) = \begin{cases} 0 & x = 0 \\ d(1, x) & 0 < x \leq L_5 \\ t_{ACK}(L_5) \cdot \left\lfloor \frac{x}{L_5} \right\rfloor + & x > L_5 \\ + t_{TX}^{DM}(x \bmod L_5) & \end{cases} \quad (4.12)$$

$$t_{TX}^{DH}(x) = \begin{cases} 0 & x = 0 \\ d(1, x) & 0 < x \leq L_{1,1} \\ d(2, x) & L_{1,1} < x \leq L_{2,1} \\ d(3, x) & L_{2,1} < x \leq L_{3,1} \\ d(2, x) & L_{3,1} < x \leq L_{2,3} \\ d(3, x) & L_{2,3} < x \leq L_{3,3} \\ d(2, x) & L_{3,3} < x \leq L_{2,5} \\ d(3, x) & L_{2,5} < x \leq L_{3,5} \\ t_{ACK}(L_{3,5}) \cdot \left\lfloor \frac{x}{L_{3,5}} \right\rfloor + & x > L_{3,5} \\ + t_{TX}^{DH}(x \bmod L_{3,5}) & \end{cases} \quad (4.13)$$

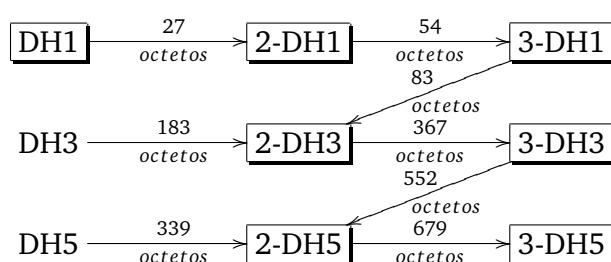
En la ecuación 4.12, L_1 , L_3 y L_5 , al igual que en la ecuación 4.11, son las longitudes máximas del *payload* de un paquete DM1, DM3 y DM5, es decir, 17, 121 y 224 octetos. Sin embargo, en la ecuación 4.13, $L_{m,s}$ denota el número de octetos de un paquete DH que ocupa s *slots* transmitido a m Mbps. La tabla 4.1 relaciona los distintos tipos de paquetes con su longitud máxima en octetos y la nomenclatura empleada en la expresión 4.13. Adicionalmente, en la figura 4.5, se representa la secuencia de uso de paquetes DH EDR formulada en la ecuación 4.13. Nótese que, conforme se incrementa el tamaño de los datos de usuario, se asume que la política propuesta de utilización de paquetes BB consiste en aumentar la velocidad de modulación antes de incrementar el número de *slots* ocupados, tal como se representa en las gráficas de la figura 4.6. Una excepción a ese comportamiento lo constituye el hecho de que el modelo excluye el uso de los paquetes DH3 y DH5, dado que los experimentos con dispositivos reales con *chipset* CSR parecen demostrar que tales paquetes no se emplean (véase el apartado 5.3).

Tanto en la ecuación 4.12 como en la 4.13, $d(m, x)$ es el tiempo de transmisión de x octetos de usuario, teniendo en cuenta el tipo de modulación empleada para transmitir el *payload* de los paquetes BB. La siguiente ecuación define su cálculo:

$$d(m, x) = \Delta_{EDR}(m) + t_{sym} \cdot n_{sym}(m, x); \quad m \in \{1, 2, 3\} \quad (4.14)$$

Tabla 4.1.: Longitudes máximas de los paquetes BB y nomenclatura asociada

Tipo	Longitud máxima (octetos)	$L_{m,s}$
DH1	27	$L_{1,1}$
DH3	183	$L_{1,3}$
DH5	339	$L_{1,5}$
2-DH1	54	$L_{2,1}$
2-DH3	367	$L_{2,3}$
2-DH5	679	$L_{2,5}$
3-DH1	83	$L_{3,1}$
3-DH3	552	$L_{3,3}$
3-DH5	1021	$L_{3,5}$

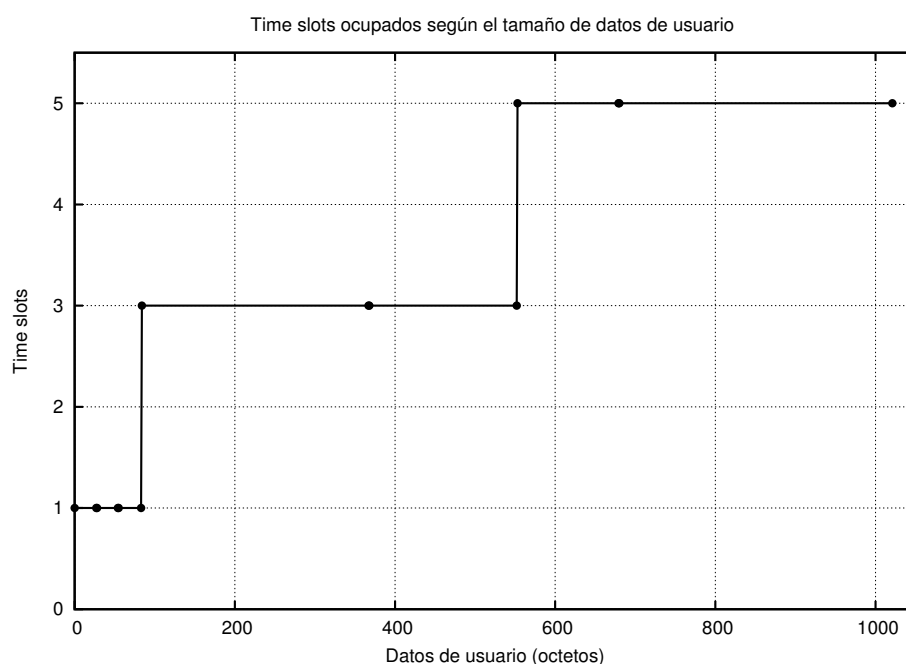
**Figura 4.5.:** Esquema de utilización de paquetes BB según se incrementa el tamaño de los datos de usuario

donde:

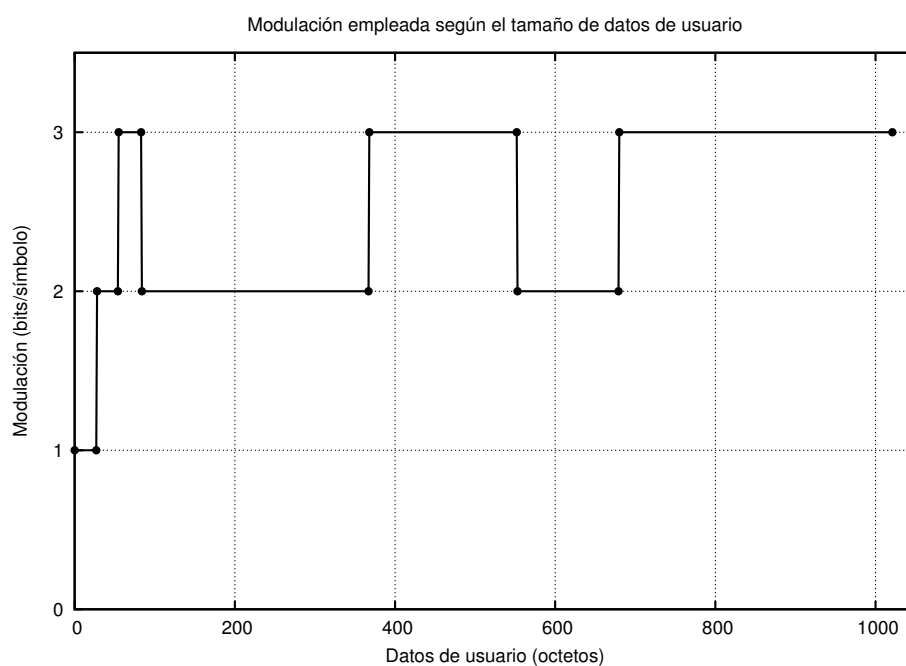
- m representa la dependencia de la modulación utilizada. Los tres posibles valores de esta variable se especifican en la tabla 4.2.

Tabla 4.2.: Definición del parámetro m en función del tipo de modulación

Modulación	Velocidad de modulación	m
GFSK	1 Mbps	1
$\pi/4$ DQPSK	2 Mbps	2
8 DPSK	3 Mbps	3



(a) Número de *time slots* ocupados en función del tamaño de datos



(b) Modulación empleada en función del tamaño de datos

Figura 4.6.: Representación del número de *time slots* ocupados y de la modulación empleada en la transmisión de x octetos de usuario, según el esquema de utilización de paquetes BB para Bluetooth 2.0 + EDR (figura 4.5).

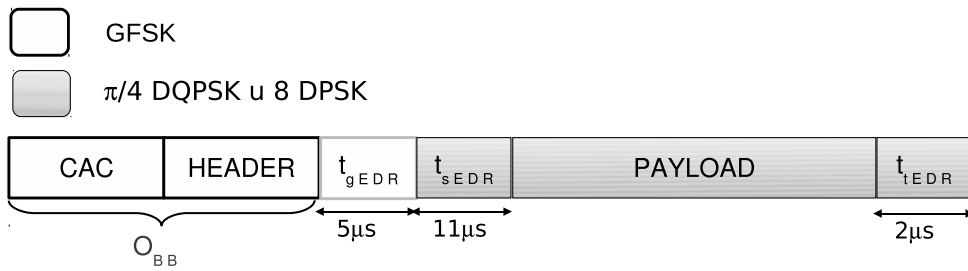


Figura 4.7.: Tiempos introducidos en un paquete Bluetooth al incorporar el modo EDR.

- $\Delta_{EDR}(m)$ es el tiempo debido a la sobrecarga añadida al *payload* de los paquetes que utilizan las modulaciones EDR. Según la norma [Bluetooth SIG04], este tiempo puede descomponerse en los siguientes sumandos, según se ha explicado en el apartado 2.2.3 y se refleja en la figura 4.7:

- **Tiempo de guarda EDR** (*Enhanced Data Rate Guard Time*) t_{gEDR} : Tiempo entre el último símbolo GFSK de la cabecera y el primer símbolo de la secuencia de sincronización EDR, cuyo valor ha de estar comprendido entre 4,75 µs y 5,25 µs. Por simplicidad, se ha considerado: $t_{gEDR} = 5 \mu s$.
- **Tiempo de la secuencia de sincronismo EDR** (*Enhanced Data Rate Synchronization Sequence*) t_{sEDR} : Consta de un símbolo de referencia DPSK (con fase arbitraria) seguido de 10 símbolos DPSK. El comienzo del símbolo de referencia debe estar sincronizado en $\pm 1/4 \mu s$ respecto al último símbolo GFSK de la cabecera. El tiempo total es: $t_{sEDR} = 11 \mu s$.
- **Tiempo de cola EDR** (*Enhanced Data Rate Trailer*) t_{tEDR} : Para finalizar la transmisión del *payload* EDR, se agregan dos símbolos EDR, correspondientes a secuencias de valor 0. Por tanto, el tiempo es: $t_{tEDR} = 2 \mu s$.

En definitiva, la sobrecarga introducida en los paquetes EDR puede obtenerse como la suma de las siguientes componentes:

$$\Delta_{EDR}(m) = \begin{cases} 0 & m = 1 \\ t_{gEDR} + t_{sEDR} + t_{tEDR} & (m = 2) \vee (m = 3) \end{cases} \quad (4.15)$$

Continuando con la definición de términos de la ecuación 4.14:

- t_{sym} es el tiempo de transmisión de un símbolo (*symbol*), $1 \mu s$, teniendo en cuenta que, con independencia de la modulación empleada, Bluetooth transmite a una velocidad de 1 Msímbolo/s.
- $n_{sym}(m, x)$, de forma equivalente a la ecuación 4.7, representa el número de símbolos (en lugar del número de bits) transmitidos en un paquete BB de x octetos. Se puede calcular mediante la siguiente ecuación:

$$n_{sym}(m, x) = O_{BB} + \left\lceil \frac{n_d(x)}{m} \right\rceil \quad (4.16)$$

siendo:

- O_{BB} es el tamaño de la sobrecarga del paquete Bluetooth, definido como la suma de los bits de la cabecera del paquete (54 bits) y del código de acceso (72 bits), lo que supone un total de 126 bits. Como esta cabecera se transmite siempre utilizando símbolos GFSK, el número de símbolos coincide con el número de bits.
- $n_d(x)$ representa el número de bits del campo de datos (*Data Field*), es decir, de la cabecera y cuerpo del *payload*, los 2 octetos de CRC y, en el caso de los paquetes DM, los bits de redundancia añadidos por la codificación FEC (véase la figura 2.4a):

$$n_d(x) = \begin{cases} (x + H_{df}^{DH}(x) + O_{CRC}) \cdot 8 & \text{paquetes DH} \\ \left\lceil \frac{(x + H_{df}^{DM}(x) + O_{CRC}) \cdot 8}{10} \right\rceil \cdot 15 & \text{paquetes DM} \end{cases} \quad (4.17)$$

En la fórmula anterior, el término O_{CRC} contabiliza la sobrecarga asociada al campo CRC, 2 octetos, mientras que $H_{df}^{DH}(x)$ y $H_{df}^{DM}(x)$ representan el número de octetos correspondientes a la cabecera del *payload*, de acuerdo con las expresiones siguientes, según se trate de paquetes DH o DM, respectivamente:

$$H_{df}^{DH}(x) = \begin{cases} 1 & x \leq L_{1,1} \\ 2 & x > L_{1,1} \end{cases} \quad (4.18)$$

$$H_{df}^{DM}(x) = \begin{cases} 1 & x \leq L_1 \\ 2 & x > L_1 \end{cases} \quad (4.19)$$

donde $L_{1,1}$, en la ecuación 4.18, se corresponde con la longitud máxima del *payload* de un paquete DH1 —27 octetos—, mientras que en el caso de los paquetes DM (4.19) el valor de L_1 es de 17 octetos. Es decir, la cabecera del *payload* es de dos octetos para todos los paquetes BB, salvo para los paquetes DH1 y DM1.

La ecuación 4.17 tiene en cuenta, asimismo, que para paquetes DM, en los que se protege la información de usuario y los dos octetos de CRC con codificación FEC $2/3$, por cada 10 bits de información se agregan 5 bits de redundancia, de forma que si el número de bits no es múltiplo de 10 hay que añadir los bits de relleno necesarios detrás del CRC.

Obsérvese que en la ecuación 4.16, con el fin de tener en cuenta el tipo de modulación, para calcular el número de símbolos del campo de datos, excluyendo los asociados al tiempo de guarda, sincronismo y cola EDR, a partir de número de bits transmitidos, se debe dividir $n_d(x)$ entre el factor m (que define el número de bits transmitidos por símbolo), redondeando al entero superior más próximo.

Cabe señalar que en las ecuaciones 4.12 y 4.13:

- Si x es menor o igual que L_5 , en el caso de paquetes DM, o de $L_{3,5}$, si se trata de paquetes DH, (la información se transporta en un único paquete BB) no se computan *slots* completos, sólo el tiempo durante el que la radio de la unidad Bluetooth destinataria está recibiendo, ya que se asume que el extremo remoto, tras la recepción del último bit, extrae los datos del paquete, los procesa para entregarlos al *Host* y, por último, emite la confirmación al origen.

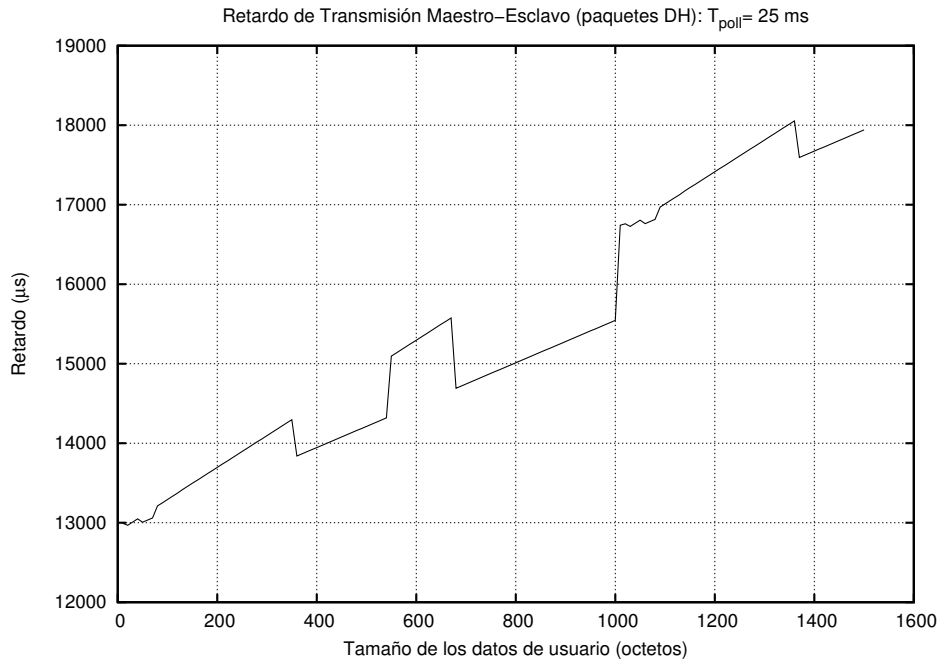


Figura 4.8.: Retardo medio calculado, en el sentido maestro-esclavo, para el perfil SPP empleando paquetes DH y considerando un intervalo de *poll* de 25 ms (40 *slots*).

- Si x es mayor que L_5 (para paquetes DM) o de $L_{3,5}$ (para paquetes DH), no se considera el tiempo de confirmación del último paquete transmitido, pero sí el de los fragmentos intermedios. De ahí que, tanto en la ecuación 4.12 como en la ecuación 4.13, se invoque la fórmula de $t_{ACK}(x)$.

En la figura 4.8 se representa el retardo medio computado, en las condiciones óptimas asumidas por el modelo, a partir de la ecuación 4.1 para el perfil SPP y paquetes DH, en el sentido maestro-esclavo y considerando un intervalo de *poll* de 25 ms (40 *slots*), valor (por defecto) que equivale a no haber configurado la calidad de servicio. Para su cálculo se han tomado los valores de MTU utilizados en la implementación de los niveles del *Host* de la arquitectura Bluetooth BlueZ [BlueZ], empleada para la validación empírica del modelo formulado (véase el capítulo 5). En esta implementación se fijan:

- La MTU de L2CAP para RFCOMM, M_R , a 1013 octetos.
- El parámetro $N1$ a 1008 octetos, decisión de implementación que permite evitar el efecto del *segmentation mismatch*, tal y como se indica en [Moron08], ya que se

satisface la igualdad: $N1 = M_R - 5$.

En la gráfica 4.8 cabe destacar que el primer escalón descendente tiene lugar cuando la longitud de los datos a transmitir, sumado a la sobrecarga añadida por cada uno de los protocolos RFCOMM y L2CAP, excede la capacidad de un paquete 2-DH3: 367 octetos. A partir de este punto, según se refleja en el esquema de la figura 4.5, se presupone que se pasa a utilizar el paquete 3-DH3 hasta que se superan los 552 octetos, valor para el que se incrementa el número de *slots*, conmutando a un paquete de menor velocidad: 2-DH5. Este paquete se continúa empleando mientras no se alcancen los 679 octetos, longitud en la que vuelve a tener lugar un segundo escalón descendente al no poder utilizar otro tipo de paquete que el de máxima capacidad y velocidad: 3-DH5. Por último, a partir de los 1008 octetos de datos de usuario, al igual que sucedía en Bluetooth 1.1, se produce un salto ascendente, ya que en el nivel RFCOMM ha de fragmentar los datos para su transporte, a fin de no violar la MTU de L2CAP.

Es conveniente aclarar que, aunque para los paquetes DM no se produce cambio alguno en las versiones posteriores a Bluetooth 1.1, en la gráfica 4.9 se ha representado el retardo para este tipo de paquetes, computado también mediante la ecuación 4.1, ya que en el capítulo 5 se incluyen los resultados de la validación empírica realizada con dispositivos conformes a Bluetooth 2.0 + EDR. Puede observarse que en el caso de utilizar paquetes DM, el retardo tiene un comportamiento creciente de forma que cada vez que se produce el llenado de un paquete DM5 tiene lugar un salto ascendente, de 6 *slots*, debido a que para mandar un paquete que transporte un fragmento no inicial hay que esperar la confirmación del anterior. Por contra, nótese que cuando se emplean paquetes DH:

- Hay escalones descendentes, ocasionados por el cambio de la modulación empleada, debido a la política de selección de paquetes indicada en la tabla 4.5.
- El incremento ascendente debido a la fragmentación en más de un paquete BB no se produce hasta que no se excede la máxima capacidad de un paquete 3-DH5 (1021 octetos).

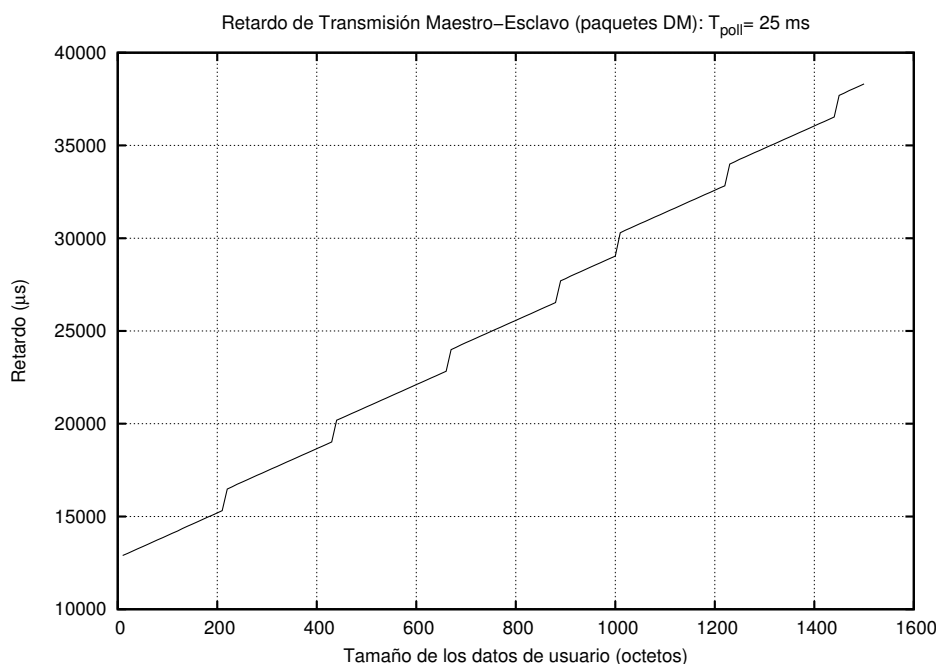


Figura 4.9.: Retardo medio calculado, en el sentido maestro-esclavo, para el perfil SPP empleando paquetes DM y considerando un intervalo de *poll* de 25 ms (40 slots).

4.4. Adaptación del modelo a Bluetooth 2.0 + EDR considerando el efecto de las retransmisiones

Como se ha podido comprobar en el apartado anterior, los retrasos máximos teóricos obtenidos para la transmisión de 1500 octetos de datos de usuario, utilizando paquetes DM, no sobrepasan los 40 ms. Este retardo, aunque es más elevado que el que se alcanza empleando paquetes DH (19 ms), no deja de ser poco relevante para la mayor parte de las aplicaciones, incluso para muchas con transmisiones en tiempo real. Además, los escenarios sin pérdidas son poco realistas, especialmente teniendo en cuenta que Bluetooth opera en la banda ISM, compartida con otras tecnologías inalámbricas. Por este motivo, el objetivo de este apartado es completar el modelo presentado en el apartado anterior, para contemplar el efecto de las retransmisiones en el cálculo del retardo asociado al perfil SPP.

En primer lugar es conveniente aclarar que se va a considerar el modo de operación básico del protocolo L2CAP (modo por defecto), en el que no se realizan retransmisiones a

nivel L2CAP y que es el que generalmente se utiliza en las aplicaciones de Bluetooth, debido a que garantiza la interoperatividad con versiones anteriores a Bluetooth 2.0 + EDR.

Por consiguiente, se trata de incorporar el efecto de las retransmisiones a nivel BB en el cálculo del retardo, lo que supone la redefinición de las ecuaciones correspondientes a las dos componentes, $t_{ACK}(x)$ y $t_{TX}(x)$. Dichas componentes que, en lo sucesivo pasan a denotarse como $\overline{t_{ACK_i}}(x)$ y $\overline{t_{TX_i}}(x)$, se calculan a partir del número medio de retransmisiones, obtenido en función de la tasa de paquetes erróneos, PER, independientemente de que se trate del modo básico o del modo EDR.

Por su parte, hay tener en cuenta que la tasa PER, tal y como se detalla en los siguientes epígrafes, se determina en función de:

- El número de bits del campo de datos (*Data Field*), $n_d(x)$, en el paquete Bluetooth (véase la ecuación 4.17).
- La tasa de bits erróneos, BER.

La ecuación 4.17, que calcula el número de bits del campo de datos, $n_d(x)$, es válida tanto para paquetes DM y DH básicos como para paquetes DH EDR. Sin embargo, para las mismas condiciones del canal de transmisión, modeladas mediante cierta relación señal/ruido, SNR, la tasa BER no es igual si emplea la modulación básica o alguna de las modulaciones EDR: $\pi/4$ DQPSK o 8 DPSK.

Asumiendo que el canal de transmisión Bluetooth se puede modelar como un canal AWGN, es decir, considerando que el efecto del ruido es equivalente a la adición de ruido gaussiano de potencia constante en la banda considerada, la tasa BER puede determinarse a partir de la relación señal/ruido, SNR.

La aplicación de este modelo para Bluetooth está avalada por su amplia utilización en trabajos relacionados, tanto para el análisis del rendimiento y como para el diseño de transceptores Bluetooth, entre los que cabe citar [Chang09, Ibrahim07, Moon07, Valenti02a]. Adicionalmente, en el estudio realizado se pueden argumentar los siguientes motivos para justificar el uso de este modelo:

- Los dispositivos Bluetooth suelen emplearse en un entorno estático de oficina, por lo que efectos como los desvanecimientos rápidos (*fading*) son poco frecuentes.

- La limitada potencia de los transmisores determina distancias de transmisión cortas, en las que suele existir línea de visión directa.
- La principal característica que define un entorno estático de oficina es la presencia de múltiples elementos interferentes en la banda ISM con diferentes características espectrales, que pueden estadísticamente modelarse como un ruido de potencia constante.

Así pues, partiendo de la premisa de que el canal de transmisión se puede modelar como un canal AWGN, en el siguiente apartado, para cada una de las modulaciones especificadas en Bluetooth, se describe cómo obtener la tasa BER en función de la relación señal/ruido, SNR.

4.4.1. Caracterización de las modulaciones digitales empleadas en Bluetooth 2.0 + EDR en un canal AWGN

Dado que el efecto de un canal AWGN es la adición de ruido de energía constante en toda la banda considerada, la relación SNR puede calcularse según se indica a continuación [Goldsmith05, cap. 6]:

$$SNR = \frac{P_r}{N_0 \cdot B} \quad (4.20)$$

siendo:

- P_r : La potencia de señal recibida a la entrada del receptor, expresada en W.
- $N_0/2$: El ruido añadido por el canal, en términos de densidad espectral de potencia (PSD, *Power Spectral Density*), en W/Hz.
- $2 \cdot B$: El ancho de banda de recepción considerado, dado en Hz.

No obstante, para permitir la comparación de distintas modulaciones digitales, es conveniente definir la relación SNR en función de la energía transmitida por bit o por símbolo:

$$SNR = \frac{P_r}{N_0 \cdot B} = \frac{E_s}{N_0 \cdot B \cdot T_s} = \frac{E_b}{N_0 \cdot B \cdot T_b} \quad (4.21)$$

donde:

- E_s y T_s : Representan, respectivamente, la energía por símbolo transmitido (en W) y el periodo de transmisión de un símbolo (en s).
- E_b y T_b : Denotan la energía por bit (en W) y el periodo de transmisión de un bit (en s), respectivamente.

De esta forma, el parámetro que se debe considerar para determinar la tasa BER ha de estar referido al cociente entre la energía por símbolo o bit, y la densidad espectral de ruido, según se indica en las siguientes ecuaciones [Goldsmith05, cap.6]:

$$\gamma_s = \frac{E_s}{N_0} \quad (4.22)$$

$$\gamma_b = \frac{E_b}{N_0} \quad (4.23)$$

donde γ_s representa la relación SNR por símbolo, mientras que γ_b denota la relación SNR por bit. La relación entre ambas se puede expresar como:

$$\gamma_b = \frac{\gamma_s}{\log_2 M} \quad (4.24)$$

donde M es el número de símbolos de la constelación de la modulación considerada.

El siguiente paso es determinar la tasa BER, expresada como probabilidad de error de bit, P_b , en función de la relación SNR γ_b para cada una de las modulaciones especificadas en Bluetooth. Previamente, es conveniente aclarar que para las constelaciones multinivel es necesario calcular la probabilidad de error de símbolo, P_s , y, posteriormente normalizarla según el número de símbolos posibles para obtener la probabilidad de error de bit, P_b . Una aproximación adecuada [Goldsmith05, cap. 5.1.5], que no considera la posibilidad de que se produzcan errores múltiples de bit por cada símbolo, viene dada por la siguiente expresión:

$$P_b \approx \frac{P_s}{\log_2 M} \quad (4.25)$$

4.4.1.1. Modulación GFSK

La modulación GFSK (*Gaussian Frequency Shift Keying*) podría asimilarse a una modulación FSK (*Frequency Shift Keying*), de forma que en caso óptimo, es decir, utilizando señales no correladas entre sí (ortogonales) y empleando un demodulador con detección coherente la probabilidad de error de bit se puede calcular como [Carlson86, cap. 14.2][Goldsmith05, cap. 6.1.5] :

$$P_b = Q(\sqrt{\gamma_b}) \quad (4.26)$$

donde $Q(z)$, que es la probabilidad de que una variable gaussiana con media 0 y varianza 1 sea mayor que z , se calcula a partir de la ecuación 4.27 o de su expresión equivalente 4.28:

$$Q(z) = p(X > z) = \int_z^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx \quad (4.27)$$

$$Q(z) = \frac{1}{2} \operatorname{erfc}\left(\frac{z}{\sqrt{2}}\right) \quad (4.28)$$

donde $\operatorname{erfc}(z)$ representa la función error complementario.

Sin embargo, debido a que Bluetooth fue concebida como tecnología de bajo coste, la opción preferida es la detección no coherente que, aunque empeora las prestaciones, permite la simplificación del demodulador. En este caso la probabilidad de error es [Carlson86, cap. 14.3]:

$$P_b = \frac{1}{2} \cdot \exp\left(-\frac{\gamma_b}{2}\right) \quad (4.29)$$

No obstante, la ecuación 4.29 representa una cota inferior, ya que, realmente, el hecho de que el estándar especifique un índice de modulación h comprendido en el intervalo $[0,28, 0,35]$ y, por tanto, menor que 0,5, implica que no se puede dar por supuesta la condición de ortogonalidad. En este caso la expresión de la probabilidad de error de bit puede calcularse como [Proakis00, cap. 5.4, eq. 53]:

$$P_b = Q_1(\alpha, \beta) - \frac{1}{2} \exp\left(-\frac{\alpha^2 + \beta^2}{2}\right) I_0(\alpha\beta) \quad (4.30)$$

donde $Q_1(a, b)$ es la función Q de Marcum, definida a partir de la siguiente ecuación:

$$\begin{aligned} Q_1(a, b) &= \int_b^{\infty} x \exp\left(-\frac{x^2 + a^2}{2}\right) I_0(ax) dx = \\ &= \exp\left(-\frac{a^2 + b^2}{2}\right) \cdot \sum_{n=0}^{\infty} \left(\frac{a}{b}\right)^n I_n(ab); \quad (b > a > 0) \end{aligned} \quad (4.31)$$

siendo $I_n(x)$ la función de Bessel modificada de orden n , y α y β , parámetros que pueden calcularse a partir de la relación SNR por bit y del denominado coeficiente de correlación ρ [Proakis00, cap. 5.4, eq. 54]:

$$\begin{aligned} \alpha &= \sqrt{\frac{\gamma_b}{2} (1 - \sqrt{1 - \rho^2})} \\ \beta &= \sqrt{\frac{\gamma_b}{2} (1 + \sqrt{1 - \rho^2})} \end{aligned} \quad (4.32)$$

obteniéndose el coeficiente de correlación ρ a partir del índice de modulación h :

$$\rho = \frac{\sin(2\pi h)}{2\pi h} \quad (4.33)$$

Así, para determinar la probabilidad de error de bit, P_b , en el modo básico de Bluetooth, hay que tener en cuenta que en este modo se emplea GFSK con índice de modulación h comprendido en el intervalo $[0,28, 0,35]$. No obstante, es conveniente aclarar que se está ignorando el efecto de la interferencia entre símbolos, ISI (*Inter-Symbol Interference*), introducida en el modulador por el filtro gaussiano.

A partir de la formulación presentada, en la gráfica de la figura 4.10 se compara la probabilidad de error correspondiente a FSK ortogonal (con detección coherente y no coherente) y GFSK no ortogonal con índices de modulación $h = 0,28$ y $h = 0,35$.

4.4.1.2. Modulación $\pi/4$ DQPSK

En cuanto al rendimiento del receptor, la modulación $\pi/4$ DQPSK (*Differential Quaternary PSK*) tiene un comportamiendo idéntico al de la modulación DQPSK (4 DPSK) en un canal

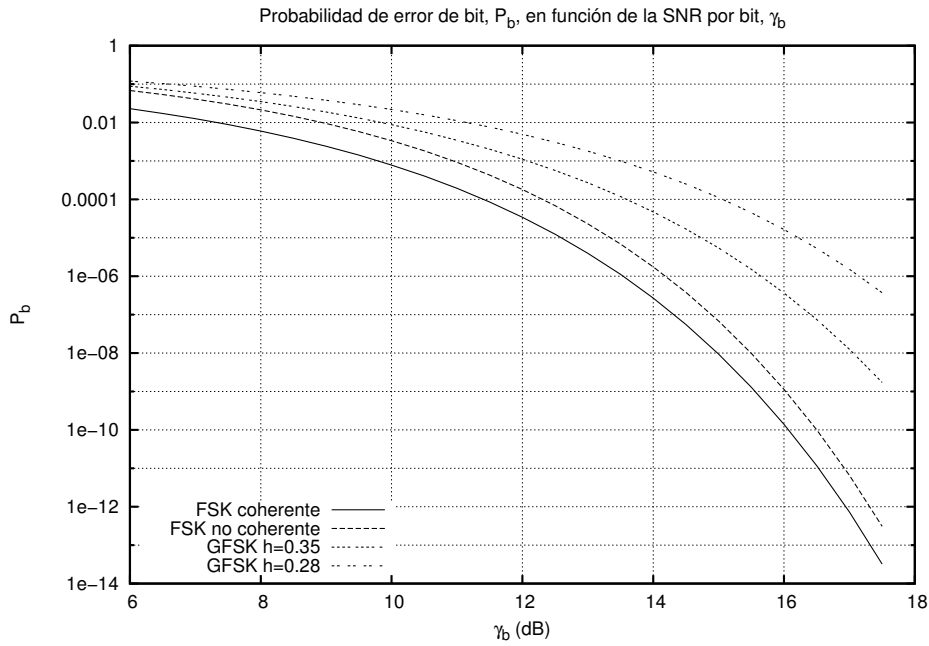


Figura 4.10.: Gráfica teórica de la probabilidad de error de bit, P_b , equivalente a la tasa BER, para las modulaciones FSK (con detección coherente y no coherente), y GFSK con índices de modulación $h = 0,28$ y $h = 0,35$, en función de la relación SNR por bit, $\gamma_b = \frac{E_b}{N_0}$.

AWGN lineal ideal. Por tanto, puede ser caracterizado por la ecuación de la probabilidad de error de símbolo, P_s , siguiente [Simon05, cap.8, eq. 84]:

$$P_s = \frac{\sin \frac{\pi}{M}}{2\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{\exp \left(\gamma_s \cdot \left(1 - \cos \frac{\pi}{M} \cdot \cos \theta \right) \right)}{1 - \cos \frac{\pi}{M} \cdot \cos \theta} d\theta \quad (4.34)$$

Para determinar la probabilidad de error de bit, P_b , hay que tener en cuenta que está relacionada con P_s según la ecuación 4.25 y, también, que se debe expresar γ_s en función de γ_b , de acuerdo con la ecuación 4.24 particularizada para M igual a 4 (ya que se codifican 2 bits por símbolo):

$$P_b = \frac{1}{\log_2 M} \cdot \frac{\sin \frac{\pi}{M}}{2\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{\exp \left(\gamma_b \cdot \log_2 M \cdot \left(1 - \cos \frac{\pi}{M} \cdot \cos \theta \right) \right)}{1 - \cos \frac{\pi}{M} \cdot \cos \theta} d\theta \quad (4.35)$$

Alternativamente, esta probabilidad puede calcularse a partir de la expresión 4.30 donde, en este caso, los parámetros α y β se calculan, a partir de la relación SNR por bit [Proakis00, cap. 5.2.8]:

$$\begin{aligned}\alpha &= \sqrt{2\gamma_b \cdot \left(1 - \sqrt{\frac{1}{2}}\right)} \\ \beta &= \sqrt{2\gamma_b \cdot \left(1 + \sqrt{\frac{1}{2}}\right)}\end{aligned}\quad (4.36)$$

parámetros que pueden aproximarse por: $\alpha \approx 0,765\sqrt{\gamma_b}$ y $\beta \approx 1,85\sqrt{\gamma_b}$ [Goldsmith05, cap 6.1.7].

La gráfica de la figura 4.11 muestra que las ecuaciones 4.35 y 4.30, con los parámetros α y β dados por la ecuación 4.36, conducen al mismo resultado.

4.4.1.3. Modulación 8 DPSK

La probabilidad de error de bit, P_b , puede determinarse también a partir de la ecuación genérica 4.35, particularizada para M igual a 8 (ya que se codifican 3 bits por símbolo). No obstante, tal y como se especifica en [Proakis00, cap. 5.2.8] el cálculo exacto de la probabilidad para la modulación M-DPSK es extremadamente complejo de realizar para valores de M distintos de 2. Por tanto, se ha estimado conveniente considerar otras cotas y aproximaciones para la probabilidad de error de símbolo, P_s , y, por ende, para P_b :

- La cota exacta facilitada por Simon [Simon05, cap.8, eq. 95] se puede computar como:

$$P_s \leq 2,06 \cdot \sqrt{\frac{1 + \cos \frac{\pi}{M}}{2 \cdot \cos \frac{\pi}{M}}} \cdot Q\left(\sqrt{\gamma_s \cdot \left(1 - \cos \frac{\pi}{M}\right)}\right) \quad (4.37)$$

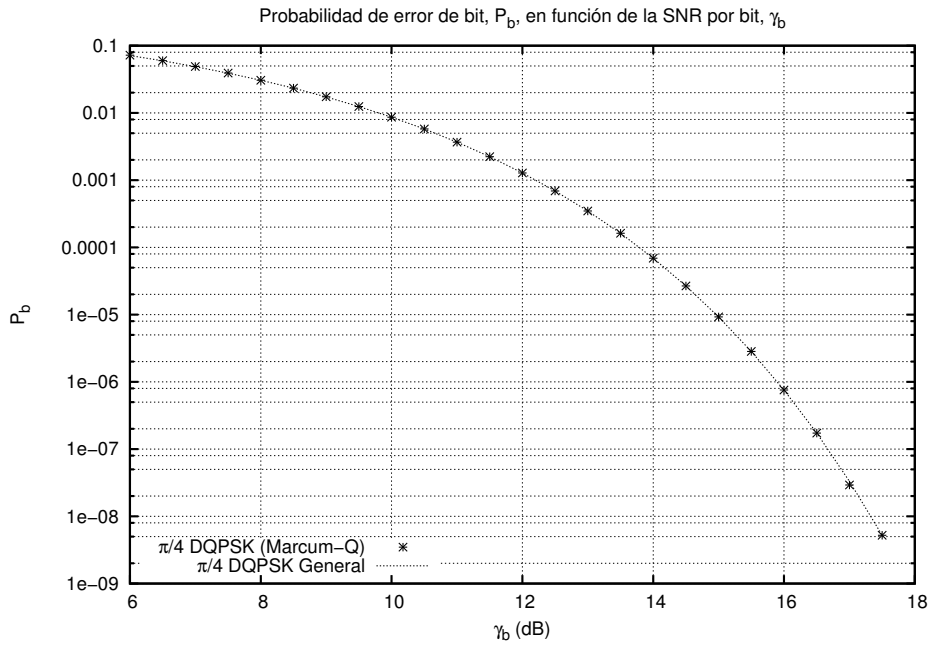


Figura 4.11.: Gráfica teórica de la probabilidad de error de bit, P_b , equivalente a la tasa BER, para la modulación $\pi/4$ DQPSK en función de la relación SNR por bit γ_b , contrastando la fórmula general 4.35 con la fórmula 4.30 (función Q de Marcum) particularizada con los parámetros dados por 4.36.

- La cota aproximada proporcionada en [Benvenuto02, cap. 6.5.1] es:

$$P_s \lesssim 1 + Q_1 \left(\sqrt{\gamma_s \cdot \left(1 - \sin \frac{\pi}{M} \right)}, \sqrt{\gamma_s \cdot \left(1 + \sin \frac{\pi}{M} \right)} \right) - Q_1 \left(\sqrt{\gamma_s \cdot \left(1 + \sin \frac{\pi}{M} \right)}, \sqrt{\gamma_s \cdot \left(1 - \sin \frac{\pi}{M} \right)} \right) \quad (4.38)$$

- La aproximación propuesta, también en [Benvenuto02, cap. 6.5.1], si M es elevada, es:

$$P_s \simeq 2 \cdot Q \left(\sqrt{\gamma_s \cdot \left(1 + \sin \frac{\pi}{M} \right)} - \sqrt{\gamma_s \cdot \left(1 - \sin \frac{\pi}{M} \right)} \right) \quad (4.39)$$

En cualquiera de los casos enumerados, la probabilidad de error de bit, P_b , se obtiene teniendo en cuenta las ecuaciones 4.25 y 4.24. Por ejemplo, su aplicación a la ecuación 4.39,

da como resultado:

$$P_b \simeq \frac{2}{\log_2 M} \cdot Q \left(\sqrt{\gamma_b \cdot \log_2 M \cdot \left(1 + \sin \frac{\pi}{M}\right)} - \sqrt{\gamma_b \cdot \log_2 M \cdot \left(1 - \sin \frac{\pi}{M}\right)} \right) \quad (4.40)$$

Expresión equivalente a la siguiente ecuación, a la que se llega mediante transformaciones trigonométricas básicas [Carlson86, cap. 14.4]:

$$P_b \simeq \frac{2}{\log_2 M} Q \left(\sqrt{4 \cdot \gamma_b \cdot \log_2 M \cdot \sin^2 \left(\frac{\pi}{2 \cdot M} \right)} \right) \quad (4.41)$$

La gráfica de la figura 4.12 ilustra que la ecuación 4.35, particularizada para $M = 8$, así como la cota definida en 4.37, y la aproximación 4.41, equivalente a la expresión 4.40, proporcionan un resultado prácticamente indistinguible.

4.4.1.4. Comparación del rendimiento de las modulaciones GFSK, $\pi/4$ DQPSK y 8 DPSK

En la figura 4.13 se representa la probabilidad de error de bit, P_b , equivalente a la tasa BER, en función de la relación SNR por bit, γ_b , expresada en dB, para las modulaciones GFSK (con índice de modulación $h = 0,35$), $\pi/4$ DQPSK y 8 DPSK. Para el cálculo de las curvas representadas, se ha utilizado la ecuación 4.30, en el caso de GFSK, y 4.35 evaluada con M igual a 4 y 8, para el caso de $\pi/4$ DQPSK y 8 DPSK, respectivamente.

En la gráfica 4.13 puede apreciarse que la modulación GFSK tiene una probabilidad de error de bit similar a la modulación $\pi/4$ DQPSK, teniendo en cuenta que el caso representado, corresponde al índice de modulación más favorable, $h = 0,35$. Este hecho justifica que la política de selección de paquetes excluya el uso de paquetes DH3 y DH5, como se ha supuesto previamente en el apartado 4.3 (véase la figura 7.1), ya que, a igualdad de robustez frente al ruido, la modulación $\pi/4$ DQPSK permite además obtener una tasa de transmisión doble. Respecto a la modulación 8 DPSK, se concluye que su rendimiento es muy inferior al

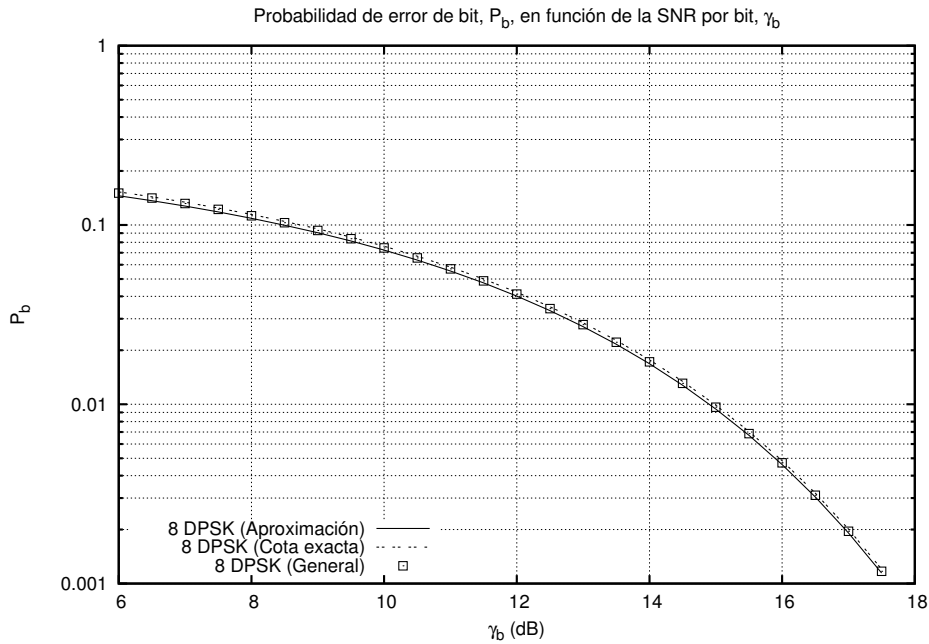


Figura 4.12.: Gráfica teórica de la probabilidad de error de bit, P_b , equivalente a la tasa BER, para la modulación 8 DQPSK en función de la relación SNR por bit, γ_b , contrastando la fórmula general 4.35 con la cota 4.37, y la aproximación 4.41, equivalente a 4.40.

obtenido con las modulaciones GFSK y $\pi/4$ DQPSK, por lo que no se considera conveniente su empleo en entornos muy ruidosos. De hecho, nótese que para una probabilidad de error de bit P_b de 10^{-3} , la modulación 8 DPSK precisa una SNR por bit γ_b en torno a 17,5 dB, mientras que las modulaciones GFSK y $\pi/4$ DQPSK, tan sólo requieren para la SNR un valor de γ_b de 12,1 dB. Por otra parte, suponiendo como valor límite de la probabilidad de error de bit P_b de 10^{-2} , a partir del cual se compromete seriamente el propio mantenimiento del enlace, se observa que en el caso de 8 DPSK la SNR debe ser superior a 15,2 dB, en contraste con los 10 dB necesarios para GFSK y $\pi/4$ DQPSK.

4.4.2. Cálculo de la tasa PER

A continuación, una vez determinada la tasa BER en función de la relación SNR por bit, γ_b , atendiendo a la modulación empleada, se propone un modelo que permite obtener

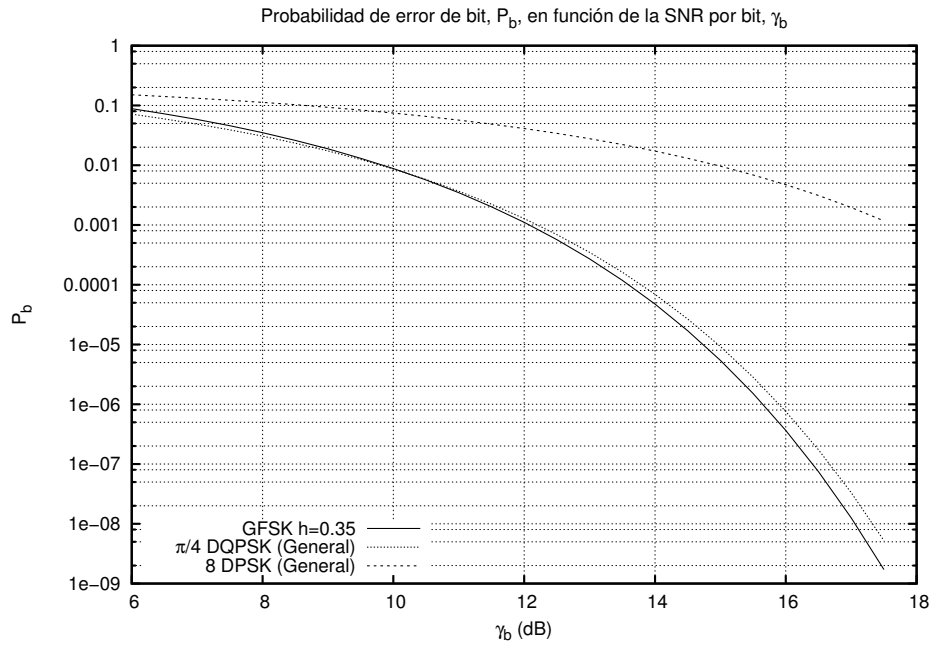


Figura 4.13.: Gráfica teórica de la probabilidad de error de bit, P_b , equivalente a la tasa BER, para las modulaciones GFSK (con índice de modulación $h = 0,35$), $\pi/4$ DQPSK y 8 DPSK, en función de la relación SNR por bit, γ_b .

las componentes $\overline{t_{ACK_i}}(x)$ y $\overline{t_{TX_i}}(x)$ del retardo de transmisión. Para ello se comienza por el cálculo de la tasa de paquetes erróneos, PER, a partir de la tasa BER.

Cálculo para el campo de datos de los paquetes DH

La probabilidad de que un paquete DH sea recibido correctamente viene dada por la probabilidad de que no exista bit erróneo alguno, dado que en dichos paquetes no se emplea ningún mecanismo de corrección. Asumiendo, por simplificar, que los errores se producen de forma incorrelada entre sí, la probabilidad de que haya i bits sin error en la transmisión de un paquete se puede caracterizar mediante una distribución binomial dada por [Papoulis02, cap.3, eq.44]:

$$P(X = i) = \binom{n}{i} \cdot p^i \cdot q^{n-i} \quad (4.42)$$

donde:

- la variable aleatoria X representa el número de bits correctos en un paquete.
- p es la probabilidad de que un bit se reciba correctamente: $p = 1 - P_b$
- $q = (1 - p)$, es la probabilidad de que un bit sea erróneo: $q = P_b$
- n simboliza el número de bits de los que consta el paquete.

Particularizando la ecuación 4.42 para $X = n_d(x)$, se obtiene la probabilidad de que un paquete DH de x octetos de carga útil sea recibido sin errores, expresada en función de la tasa BER, $P_{DH_d}(x)$, independientemente de que se trate del modo básico o EDR:

$$P_{DH_d}(x) = P(X = n_d(x)) = (1 - P_b)^{n_d(x)} \quad (4.43)$$

Cálculo para el campo de datos de los paquetes DM

En el caso de los paquetes DM, en los que se utiliza, como mecanismo de detección y corrección de errores, codificación FEC^{2/3}, hay que tener en cuenta que se trata de un código de Hamming (15,10) recortado (*shortened*), en el que por cada 10 bits se añaden 5 bits de redundancia, de forma que un bloque o grupo de 15 bits se considera correcto si hay un bit erróneo como máximo, ya que el código permite la recuperación de un error en cada grupo. Por consiguiente, la probabilidad de que no se produzca un error irrecuperable en un grupo, P_g , vendrá dada por la probabilidad de que sean correctos al menos 14 bits:

$$P_g = \sum_{i=14}^{15} \binom{15}{i} \cdot (1 - P_b)^i \cdot P_b^{(15-i)} \quad (4.44)$$

Así, la pérdida de un paquete DM se producirá cuando tengan lugar dos o más errores en, al menos, un bloque. En consecuencia, la probabilidad de éxito, es decir de que no se pierda un paquete DM, $P_{DM_d}(x)$, expresada en función de la probabilidad P_g , definida por la ecuación 4.44, es la probabilidad de que no se pierda ninguno de los grupos de 15 bits que lo componen:

$$P_{DM_d}(x) = P(X = n_g(x)) = (P_g)^{n_g(x)} \quad (4.45)$$

donde $n_g(x) = n_d(x)/15$, representa el número de grupos de 15 bits de los que consta el paquete DM, y $n_d(x)$ se obtiene a partir de la segunda rama de la ecuación 4.17, que tiene en cuenta que, además del campo de datos, se protege la cabecera del *payload* y los dos octetos de CRC.

Finalmente, sustituyendo 4.44 en la ecuación 4.45, resulta:

$$P_{DM_d}(x) = \left(15 \cdot (1 - P_b)^{(15-1)} \cdot P_b + (1 - P_b)^{15}\right)^{n_g(x)} \quad (4.46)$$

Es conveniente aclarar que se considera despreciable la posibilidad de que se produzcan errores no detectados a nivel BB dado que, tanto en los paquetes DM como en los paquetes DH, se incluye un código de redundancia cíclica de 16 bits (CRC-16), con polinomio generador $g(D) = (D + 1) \cdot (D^7 + D^4 + D^3 + D^2 + 1)$, capaz de detectar todos los errores simples, por ser $g(D)$ distinto del polinomio 1, y todos los errores múltiples impares, por ser $g(D)$ múltiplo de $(D + 1)$ [Moon05]. Por consiguiente, el modelo asume que todos los paquetes erróneos son detectados y, por tanto, retransmitidos.

Cálculo para la cabecera de los paquetes BB

La cabecera de los paquetes Bluetooth se protege con codificación FEC $1/3$, de forma que cada uno de los 18 bits de los que consta la cabecera sin codificar se transforman en 3 bits, resultando un total de 54 bits, o lo que es lo mismo, 18 grupos de 3 bits.

Teniendo en cuenta que la protección utilizada permite corregir hasta 1 bit en cada bloque de 3 bits, para determinar la probabilidad de que no se produzca un error irrecuperable en la cabecera, P_H , se ha de seguir el mismo procedimiento por el que se ha obtenido la ecuación 4.46. Así, esta probabilidad vendrá dada por la ecuación:

$$P_H = \left(3 \cdot (1 - P_b)^2 \cdot P_b + (1 - P_b)^3\right)^{18} \quad (4.47)$$

Cálculo de la probabilidad de error de un paquete completo

La probabilidad de que no se pierda un paquete, $P_{nl}(x)$, se computa a partir de la probabilidad de que no se produzca un error irrecuperable ni en la cabecera ni en el campo de datos:

$$P_{nl}(x) = \begin{cases} P_{DH_d}(x) \cdot P_H & \text{paquetes DH} \\ P_{DM_d}(x) \cdot P_H & \text{paquetes DM} \end{cases} \quad (4.48)$$

Nótese que la ecuación anterior calcula la probabilidad de la intersección de dos sucesos A y B independientes, $p(A \cap B)$, ya que se asume que no hay correlación entre los errores de bit. Por tanto se cumple que:

$$p(A \cap B) = p(A) \cdot p(B) \quad (4.49)$$

Por consiguiente, la probabilidad de que se pierda un paquete, $P_l(x)$, vendrá dada por la siguiente ecuación:

$$P_l(x) = 1 - P_{nl}(x) \quad (4.50)$$

4.4.3. Cálculo del retardo de transmisión

Asumiendo que el temporizador *flushTO* se configura para realizar infinitas retransmisiones, valor por defecto (véase el apartado 2.2.1.4), el número medio de intentos necesarios para que un paquete sea transmitido con éxito, denotado por $\overline{N_{Tx}}$, se determina a partir de la siguiente ecuación:

$$\begin{aligned} \overline{N_{Tx}} &= 1 \cdot (1 - P_l) + 2 \cdot P_l \cdot (1 - P_l) + 3 \cdot P_l \cdot P_l \cdot (1 - P_l) + \dots = \\ &= (1 - P_l) \cdot \sum_{i=1}^{\infty} i \cdot P_l^{(i-1)} = \\ &= \frac{1}{1 - P_l} \end{aligned} \quad (4.51)$$

Asimismo, $\overline{t_{ACK_l}}(x)$, es decir, el tiempo medio de transmisión y confirmación de un paquete, teniendo en cuenta el número medio de transmisiones que se realizan y la confirmación del último intento realizado satisfactoriamente, viene dado, para valores de x inferiores a la máxima capacidad de un paquete de 5 slots, por la siguiente ecuación:

$$\begin{aligned}\overline{t_{ACK_l}}(x) &= \frac{P_l}{1 - P_l} \cdot t_{RTx}(x) + t_{ACK}(x) \\ &= (\overline{N_{Tx}} - 1) \cdot t_{RTx}(x) + t_{ACK}(x)\end{aligned}\quad (4.52)$$

Donde, obviamente, $(\overline{N_{Tx}} - 1)$ representa el número de veces que un paquete es retransmitido.

Al igual que en el medio sin pérdidas, considerando que la longitud de los datos pueda exceder la máxima capacidad de un paquete de 5 slots, el retardo $\overline{t_{ACK_l}}(x)$ se calcularía de forma recursiva como:

$$\overline{t_{ACK_l}}(x) = \begin{cases} 0 & x = 0 \\ (\overline{N_{Tx}} - 1) \cdot t_{RTx}(x) + t_{ACK}(x) & 0 < x \leq L_5 \\ ((\overline{N_{Tx}} - 1) \cdot t_{RTx}(L_5) + t_{ACK}(L_5)) \cdot \left\lfloor \frac{x}{L_5} \right\rfloor + \overline{t_{ACK_l}}(x \bmod L_5) & x > L_5 \end{cases} \quad (4.53)$$

El retardo provocado por cada retransmisión, notado por $t_{RTx}(x)$, se ha de encontrar en el intervalo:

$$t_{RTx_{min}}(x) \leq t_{RTx}(x) \leq t_{RTx_{max}}(x) \quad (4.54)$$

donde los extremos del intervalo, $t_{RTx_{min}}(x)$ y $t_{RTx_{max}}(x)$ vienen dados por las siguientes ecuaciones:

$$t_{RTx_{min}}(x) = \begin{cases} 0 & x = 0 \\ 2 \cdot T_s & 0 < x \leq L_1 \\ 4 \cdot T_s & L_1 < x \leq L_3 \\ 6 \cdot T_s & L_3 < x \leq L_5 \end{cases} \quad (4.55)$$

$$t_{RTx_{max}}(x) = T_{poll} \quad (4.56)$$

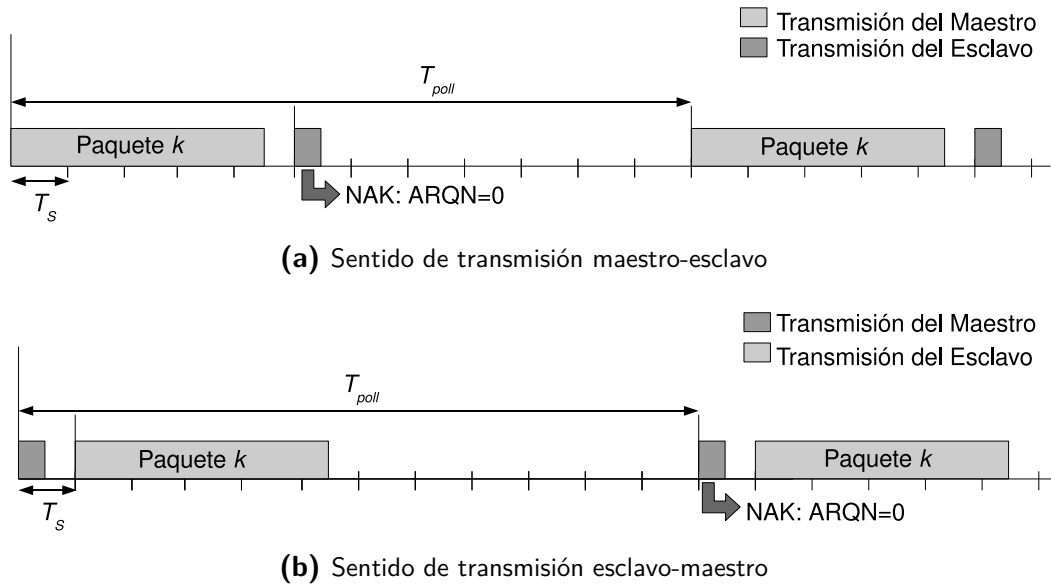


Figura 4.14.: Tiempo transcurrido desde que se inicia la transmisión de un paquete de 5 slots hasta que se realiza la retransmisión, particularizando $t_{RTx}(N)$ al valor máximo, $t_{RTx_{max}}(N)$ (fuente: [Moron08])

En la ecuación 4.55, al igual que en la ecuación 4.11, L_1 , L_3 y L_5 corresponden a las longitudes máximas del *payload* de un paquete del nivel BB de 1, 3 ó 5 slots, que serán, respectivamente, 17, 121 y 224 octetos para los paquetes DM1, DM3 y DM5, y 83, 552 y 1021 octetos para los paquetes de máxima velocidad (3 Mbps) 3-DH1, 3-DH3 y 3-DH5.

Puede observarse que la fórmula de $t_{RTx_{max}}(x)$ tiene en cuenta que los fragmentos intermedios de una PDU de L2CAP podrían estar separados hasta un máximo de T_{poll} segundos (n_{poll} slots). La figura 4.14 ilustra el significado de $t_{RTx_{max}}(x)$ para cada uno de los sentidos de comunicación, en el caso de la retransmisión de un paquete de 5 slots. Nótese que la figura 4.14a ilustra el caso límite en el que el maestro se demora hasta el máximo permitido (T_{poll}) para retransmitir al esclavo el paquete recibido por éste erróneamente. En el sentido opuesto, según se muestra en la figura 4.14b, el maestro tarda T_{poll} segundos en notificar al esclavo que el paquete no ha sido recibido correctamente, de forma que el esclavo lo retransmite a partir del siguiente slot.

Sin embargo, el extremo inferior del intervalo, $t_{RTx_{min}}(x)$, definido por la ecuación 4.55, considera, tal y como se observa en la figura 4.15 que, para el sentido de transmisión esclavo-maestro (4.15b), el maestro solicita la retransmisión inmediatamente y que, en el

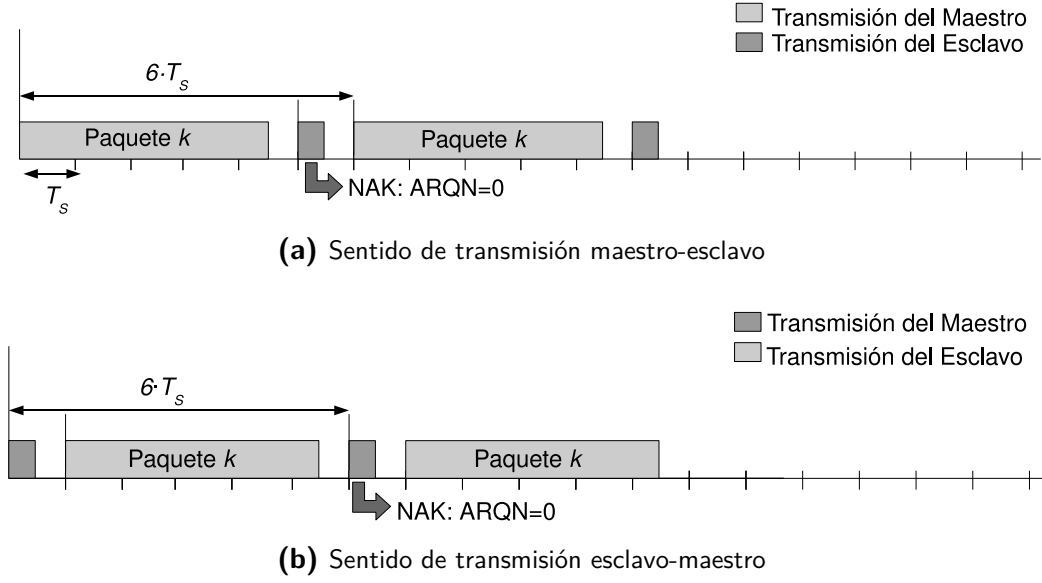


Figura 4.15.: Tiempo transcurrido desde que se inicia la transmisión de un paquete de 5 *slots* hasta que se realiza la retransmisión, particularizando $t_{RTx}(N)$ al valor mínimo, $t_{RTx_{min}}(N)$ (fuente: [Moron08])

sentido maestro-esclavo (4.15a), una vez que el maestro recibe la confirmación negativa del esclavo, procede a la retransmisión en el siguiente *slot*. Así, el retardo de retransmisión $t_{RTx_{min}}(x)$ viene determinado por el tamaño del paquete perdido (1, 3 ó 5 *slots*) más el *slot* empleado en la confirmación negativa (*NAK*).

Finalmente, siguiendo el procedimiento indicado para $\overline{t_{ACK_l}}(x)$, la definición del tiempo medio de transmisión, $\overline{t_{TX_l}}(x)$, considerando las retransmisiones realizadas, responde a la siguiente ecuación:

$$\overline{t_{TX_l}}(x) = \begin{cases} 0 & x = 0 \\ (\overline{N_{Tx}} - 1) \cdot t_{RTx}(x) + t_{TX}(x) & 0 < x \leq L_5 \\ \overline{t_{ACK_l}}(L_5) \cdot \left\lfloor \frac{x}{L_5} \right\rfloor + \overline{t_{TX_l}}(x \bmod L_5) & x > L_5 \end{cases} \quad (4.57)$$

La expresión anterior tiene en cuenta las retransmisiones (mediante la inclusión de $\overline{t_{ACK_l}}(x)$) de los paquetes no finales ya que no se va a transmitir el siguiente paquete hasta

no haber recibido la confirmación del actual.

Por consiguiente, la inclusión de 4.53 y 4.57 en la ecuación de retardo 4.1 permite definir $\overline{t_{R_{M_l}}}(N)$, el retardo medio para el perfil SPP en el sentido de transmisión maestro-esclavo, teniendo en cuenta las retransmisiones:

$$\overline{t_{R_{M_l}}}(N) = \frac{T_{poll}}{2} + n_{nf} \cdot \overline{t_{ACK_l}}(M(L_R)) + \overline{t_{TX_l}}(M(L_{ff}(N))) \quad (4.58)$$

En el sentido de comunicación esclavo-maestro la cota $\overline{t_{R_{E_l}}}(N)$ se obtiene a partir de la definida para el maestro con tan solo sumar T_S , para incluir el *slot* en el que el esclavo es direccionado por el maestro:

$$\overline{t_{R_{S_l}}}(N) = T_S + \overline{t_{R_{M_l}}}(N) \quad (4.59)$$

Es conveniente aclarar que las ecuaciones 4.58 y 4.59 permiten obtener cotas del retardo medio cuando se particulariza el cálculo de $\overline{t_{ACK_l}}(x)$ y $\overline{t_{TX_l}}(x)$ asumiendo que $t_{RTx}(x)$ se computa con la fórmula del valor máximo: $t_{RTx_{max}}(x)$.

4.4.4. Análisis de la aplicación del modelo

Para estudiar las consecuencias prácticas del modelo, se analizan los resultados que se obtienen con su aplicación bajo diversas condiciones de relación SNR, considerando dos tamaños de datos a transmitir: 70 y 200 octetos. Al respecto, es conveniente aclarar que se ha elegido el tamaño de 200 octetos teniendo en cuenta que el límite para comparar entre sí los distintos tipos de paquetes viene impuesto por el máximo número de octetos que pueden enviarse en el paquete de mínima capacidad, 224 octetos correspondientes a un paquete DM5. Nótese que, por encima de este valor, la comparación no sería justa debido a que para la transmisión de longitudes superiores, se podría requerir fragmentar los datos en más de un paquete, dependiendo del tipo de paquete considerado.

En cuanto al retardo obtenido para paquetes DH del modo EDR, cabe destacar que cuando se utiliza la modulación 8DPSK el retardo es muy elevado al tener un

comportamiento mucho peor que el resto de las modulaciones para una determinada relación SNR. La tabla 4.3 ilustra este comportamiento, indicando, en condiciones ideales ($\gamma_b = \infty$) y para una relación SNR de $\gamma_b = 10,5$ dB, $\gamma_b = 12$ dB y $\gamma_b = 15$ dB, las tasas BER y PER y el retardo de transmisión $\overline{t_{TX_i}}(M(N))$, en función del tipo de paquete: DM, DH básico o DH EDR. Dicho retardo se ha obtenido a partir de la ecuación 4.57, para el número de octetos especificados en la primera columna, N , y considerando la correspondiente sobrecarga agregada por los protocolos RFCOMM y L2CAP, $M(N)$.

Puede observarse que, por ejemplo, cuando la relación SNR es de 10,5 dB, transmitir 200 octetos de usuario empleando un paquete DM5 sólo supone 9 ms, mientras que con un paquete DH5 o 2-DH3, teóricamente, el retardo aumentaría hasta 136 s ó 143 s, valores muy elevados que, en la realidad, no se llegarían a alcanzar porque antes el nivel BB asumiría que se ha producido la pérdida del enlace.

Nótese también que, para la misma relación SNR, la transmisión de 70 octetos en un paquete 3-DH1 sería inviable y, aunque al utilizar un paquete DH3 el tiempo resulta estar acotado a 302 ms, valor que puede considerarse tolerable, contrasta con los tan solo 2,9 ms que se requieren usando un paquete DM3.

Al aumentar la relación SNR hasta 12 dB, pese a que continúa no siendo factible emplear un paquete 3-DH1 para enviar 70 octetos, sí es posible transmitir 200 octetos de usuario en un paquete DH5 o 2-DH3 en un tiempo razonable de 50 ms ó 60 ms, respectivamente. No obstante, sigue siendo preferible optar por un paquete DM5, ya que se necesitan únicamente 2,9 ms, prácticamente igual que cuando la relación SNR empeora hasta los 10,5 dB.

Obsérvese que es necesario aumentar la relación SNR hasta 15 dB para que el tiempo de transmisión consumido al utilizar un paquete 2-DH3 sea equiparable, e incluso ventajoso, frente al uso de un paquete DM5 o DH5.

Adicionalmente, cabe señalar que, aunque para 15 dB de SNR ya es viable el uso de paquetes 3-DH1, no es aconsejable emplearlos porque se requieren 3,5 s para enviar 70 octetos, cuando, de optar por el modo básico, se reducen los tiempos a 1,1 ms (con DM3) y a 0,8 ms (con DH3).

Las gráficas de la figura 4.16 representan el retardo de transmisión $\overline{t_{TX_i}}(M(N))$ de 70 y 200 octetos de datos de usuario, respectivamente, respecto a la variación de la relación

Tabla 4.3.: Comparación de la tasa BER, la tasa PER y el retardo medio obtenidos en condiciones ideales y para una relación SNR de 10,5 dB, 12 dB y 15 dB, y 70 y 200 octetos de datos de usuario, en función del tipo de paquete empleado.

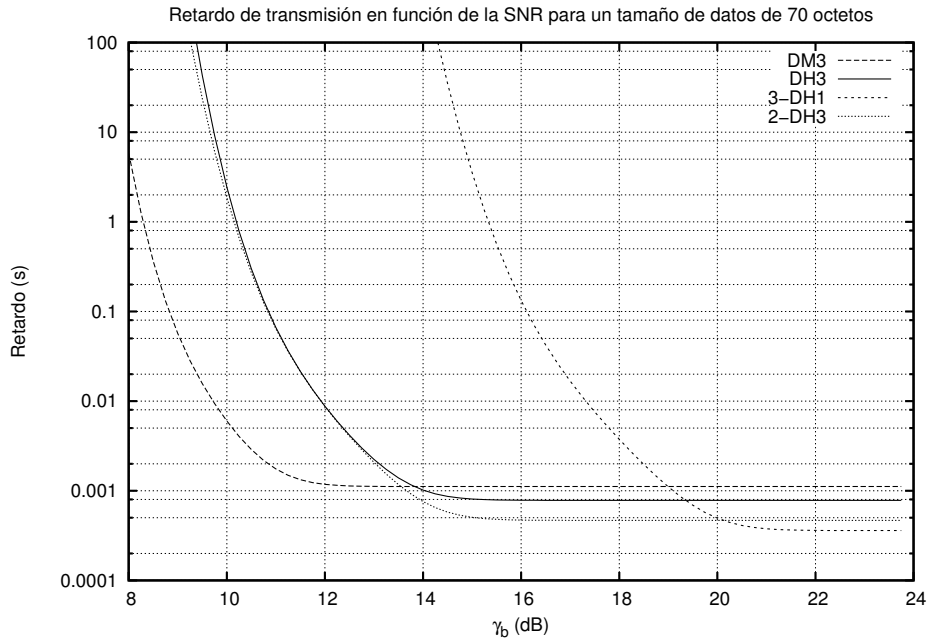
γ_b (dB)	N (octetos)	BER	Tipo de paquete	PER	$\overline{t_{TX_i}}(M(N))(s)$
∞	70	0	DM3	0	$1,116 \cdot 10^{-3}$
∞	70	0	DH3	0	$7,820 \cdot 10^{-4}$
∞	70	0	3-DH1	0	$3,600 \cdot 10^{-4}$
∞	200	0	DM5	0	$2,691 \cdot 10^{-3}$
∞	200	0	DH5	0	$1,830 \cdot 10^{-3}$
∞	200	0	2-DH3	0	$9,960 \cdot 10^{-4}$
15	70	$5,426 \cdot 10^{-6}$	DM3	$2,056 \cdot 10^{-7}$	$1,116 \cdot 10^{-3}$
15	70	$5,426 \cdot 10^{-6}$	DH3	$3,553 \cdot 10^{-3}$	$8,087 \cdot 10^{-4}$
15	70	$9,632 \cdot 10^{-3}$	3-DH1	$9,983 \cdot 10^{-1}$	3,570
15	200	$5,426 \cdot 10^{-6}$	DM5	$5,301 \cdot 10^{-7}$	$2,691 \cdot 10^{-3}$
15	200	$5,426 \cdot 10^{-6}$	DH5	$9,203 \cdot 10^{-3}$	$1,911 \cdot 10^{-3}$
15	200	$9,264 \cdot 10^{-6}$	2-DH3	$1,566 \cdot 10^{-2}$	$1,115 \cdot 10^{-3}$
12	70	$1,106 \cdot 10^{-3}$	DM3	$8,432 \cdot 10^{-3}$	$1,180 \cdot 10^{-3}$
12	70	$1,106 \cdot 10^{-3}$	DH3	$5,163 \cdot 10^{-1}$	$8,786 \cdot 10^{-3}$
12	70	$4,117 \cdot 10^{-2}$	3-DH1	$\simeq 1,000$	∞
12	200	$1,106 \cdot 10^{-3}$	DM5	$2,160 \cdot 10^{-2}$	$2,884 \cdot 10^{-3}$
12	200	$1,106 \cdot 10^{-3}$	DH5	$8,484 \cdot 10^{-1}$	$5,079 \cdot 10^{-2}$
12	200	$1,282 \cdot 10^{-3}$	2-DH3	$8,877 \cdot 10^{-1}$	$6,030 \cdot 10^{-2}$
10,5	70	$5,648 \cdot 10^{-3}$	DM3	$1,915 \cdot 10^{-1}$	$2,892 \cdot 10^{-3}$
10,5	70	$5,648 \cdot 10^{-3}$	DH3	$9,757 \cdot 10^{-1}$	$3,019 \cdot 10^{-1}$
10,5	70	$6,554 \cdot 10^{-2}$	3-DH1	$\simeq 1,000$	∞
10,5	200	$5,648 \cdot 10^{-3}$	DM5	$4,219 \cdot 10^{-1}$	$9,076 \cdot 10^{-3}$
10,5	200	$5,648 \cdot 10^{-3}$	DH5	$\simeq 9,999 \cdot 10^{-1}$	136,2
10,5	200	$5,767 \cdot 10^{-3}$	2-DH3	$\simeq 9,999 \cdot 10^{-1}$	143,1

SNR por bit, γ_b . El cálculo del retardo se ha realizado considerando el empleo de paquetes DM, DH, y EDR, de duración adecuada al tamaño de datos enviado y, en el caso de EDR, seleccionándolos según la política reflejada en la figura 4.5. Adicionalmente, en la figura 4.16 se han representado las curvas del retardo de transmisión en el caso de utilizar la modulación alternativa a la correspondiente según el esquema de selección propuesto, a fin de mostrar que:

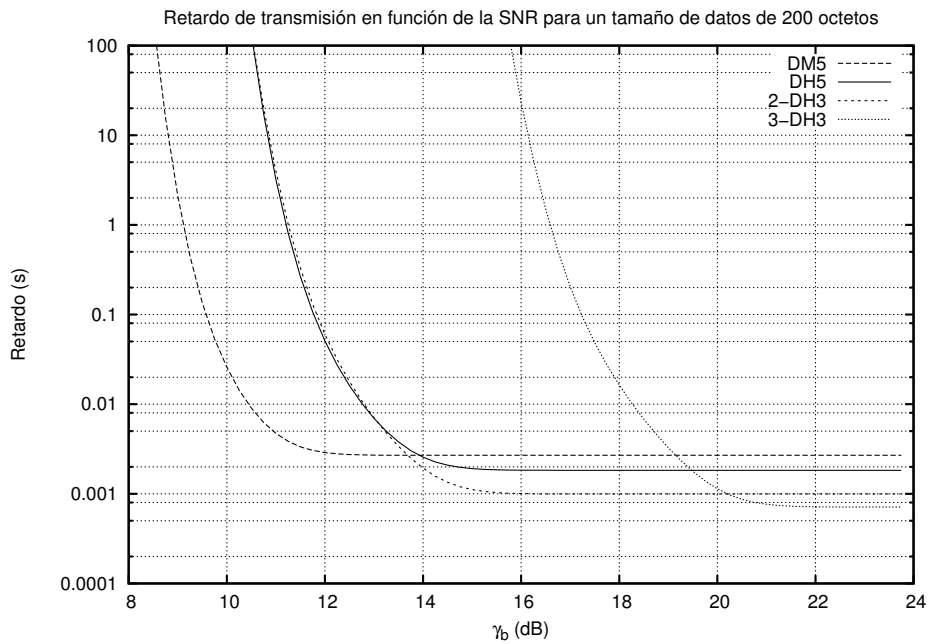
- Para la transmisión de 70 octetos de datos de usuario, el paquete utilizado siguiendo el esquema de la figura 4.5, 3-DH1, no es el idóneo si se producen pérdidas en la transmisión. En este caso resultaría apropiado el empleo de un paquete 2-DH3.
- Para la transmisión de 200 octetos de datos de usuario, el uso de un paquete 2-DH3, según la política de selección seguida, sería adecuado en un escenario con pérdidas, resultando desaconsejable optar por un paquete 3-DH3.

A modo de conclusión, del análisis realizado se pueden inferir criterios de decisión sobre la utilización de las diferentes modulaciones y tipos de paquetes, considerando la política reflejada en la figura 4.5, para longitudes inferiores a la máxima capacidad de un paquete DM5, 224 octetos, y los siguientes intervalos de la relación SNR:

- $\gamma_b \leq 8,5$ dB: No es viable mantener el enlace, independientemente del tipo de paquete empleado.
- $8,5 < \gamma_b \leq 10$ dB: Sólo es factible emplear paquetes DM, ya que el uso de los paquetes DH del modo básico y EDR conlleva retardos intolerables o, incluso, la pérdida del enlace.
- $10 < \gamma_b \leq 13,8$ dB: Es aceptable la utilización de paquetes DH y 2-DH, aunque entre ambos, 2-DH es más eficiente a partir de 13 dB. Sin embargo, sigue siendo óptima la transmisión de información en paquetes DM.
- $13,8 < \gamma_b \leq 19$ dB: Es óptima la utilización de paquetes 2-DH y, en su defecto, es admisible optar por paquetes DH del modo básico. Únicamente, a partir de 16 dB, se puede considerar el uso de paquetes 3-DH.



(a) Transmisión de 70 octetos, utilizando paquetes DM3, DH3, 3-DH1 y 2-DH3



(b) Transmisión de 200 octetos, utilizando paquetes DM5, DH5, 2-DH3 y 3-DH3

Figura 4.16.: Retardo de transmisión $\overline{t_{TXl}}(M(N))$, en función de la relación SNR por bit, γ_b , utilizando paquetes del modo básico y EDR

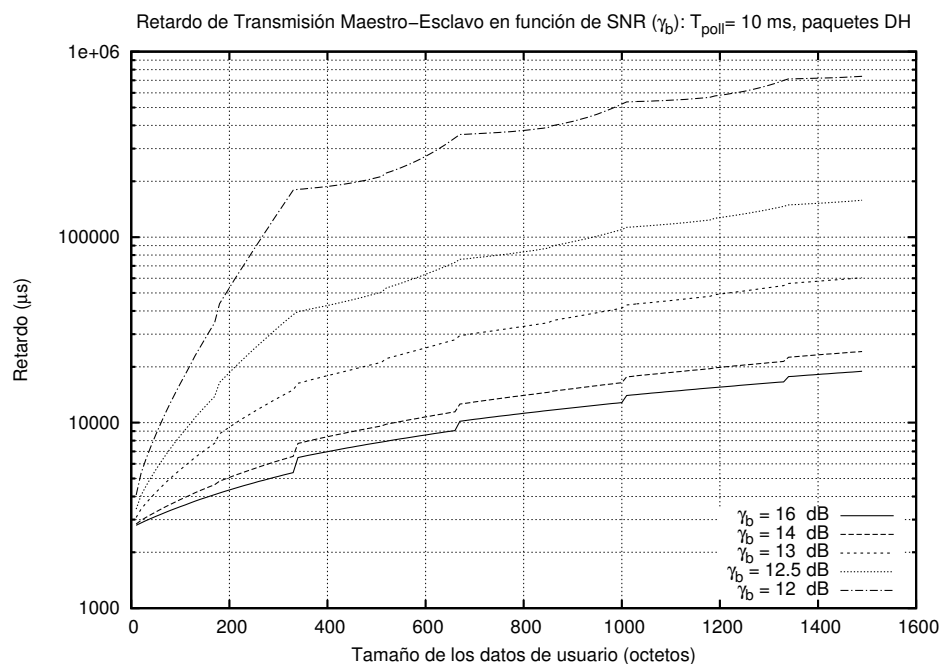
Tabla 4.4.: Tasas BER, para GFSK, correspondientes a las relaciones SNR empleadas en la figura 4.17.

Paquetes DH		Paquetes DM	
SNR, γ_b (dB)	BER	SNR, γ_b (dB)	BER
16	$3,65 \cdot 10^{-7}$	12	$1,11 \cdot 10^{-3}$
14	$4,73 \cdot 10^{-5}$	11	$3,47 \cdot 10^{-3}$
13	$2,71 \cdot 10^{-4}$	10,5	$5,65 \cdot 10^{-3}$
12,5	$5,68 \cdot 10^{-4}$	10	$8,78 \cdot 10^{-3}$
12	$1,11 \cdot 10^{-3}$	9,5	$1,31 \cdot 10^{-2}$

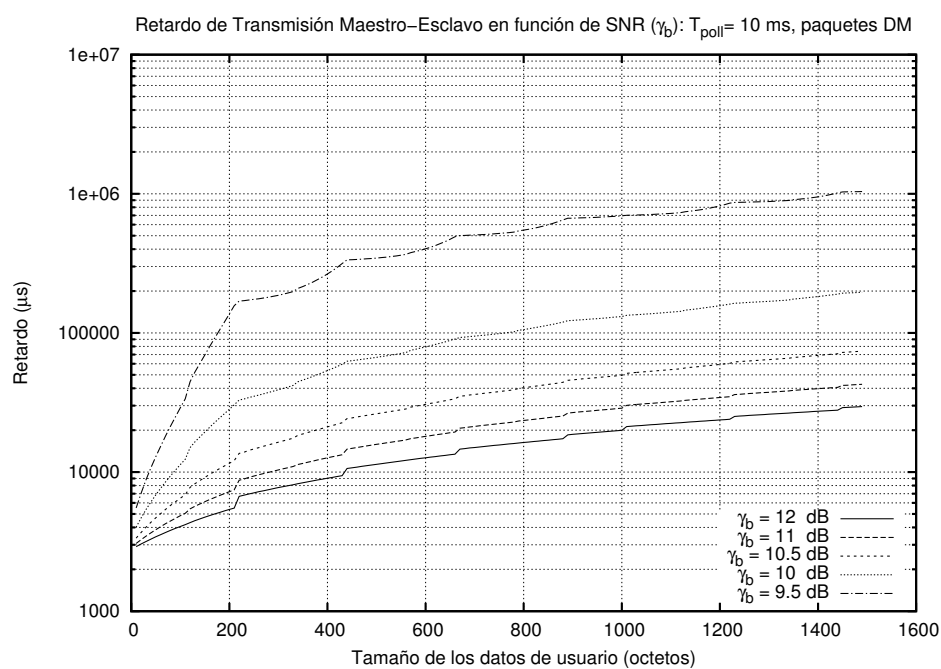
- $\gamma_b > 19$ dB: A partir de este límite puede empezar a considerarse óptima la transmisión de información empleando paquetes 3-DH.

Finalmente, en la figura 4.17 se representan gráficamente, para el perfil SPP, los retardos máximos teóricos para un valor de T_{poll} de 10 ms, utilizando tanto paquetes DM como DH del modo básico, para el sentido de transmisión maestro-esclavo. Los retardos son los proporcionados por la ecuación 4.58 para los valores de tasa BER calculados a partir de la ecuación 4.30, correspondiente a la modulación GFSK, particularizada para un índice de modulación de 0,35 ($h = 0,35$) y las relaciones SNR indicadas en la tabla 4.4. Cabe destacar que en el caso de emplear paquetes DH, para una relación SNR de 12 dB, $\gamma_b = 12$ dB, que tiene asociada una tasa BER del 0,11 %, se alcanzan retardos cercanos a 1 s, orden de magnitud que, para los paquetes DM, corresponde a una relación SNR de 9,5 dB, $\gamma_b = 9,5$ dB y a una tasa BER del 1,31 %. En definitiva, estos resultados ponen de manifiesto la utilidad de los paquetes DM, con codificación FEC $2/3$, ya que permiten operar, incluso, con tasas BER superiores al 1 %.

En lo que respecta al modo EDR, en la figura 4.18 se han representado las curvas de retardo de transmisión para el perfil SPP obtenidas también a partir de la ecuación 4.58 para los valores de relación SNR comprendidos entre 21 y 18 dB. Nótese que, en este caso, a diferencia del modo básico, no es factible indicar la tasa BER asociada a cada valor de la relación SNR, dado que la cabecera del paquete se transmite siempre utilizando la modulación GFSK, mientras que la modulación empleada para el campo de datos ($\pi/4$ DQPSK u 8 DPSK) varía con el número de octetos a transportar, de acuerdo con



(a) Paquetes DH (modo básico)



(b) Paquetes DM

Figura 4.17.: Efecto de la relación SNR en las transmisiones maestro-esclavo para el perfil SPP.

la política de selección esquematizada en la figura 4.5. Estas gráficas ilustran la acusada diferencia existente entre los valores calculados para paquetes 3-DH y los obtenidos para otras modulaciones, ya que, como se ha explicado, el empleo de paquetes 3-DH sólo empieza a ser planteable para una relación SNR $\gamma_b = 19$ dB, que corresponde a una tasa BER inferior al 0,02 % para la modulación empleada, 8 DPSK.

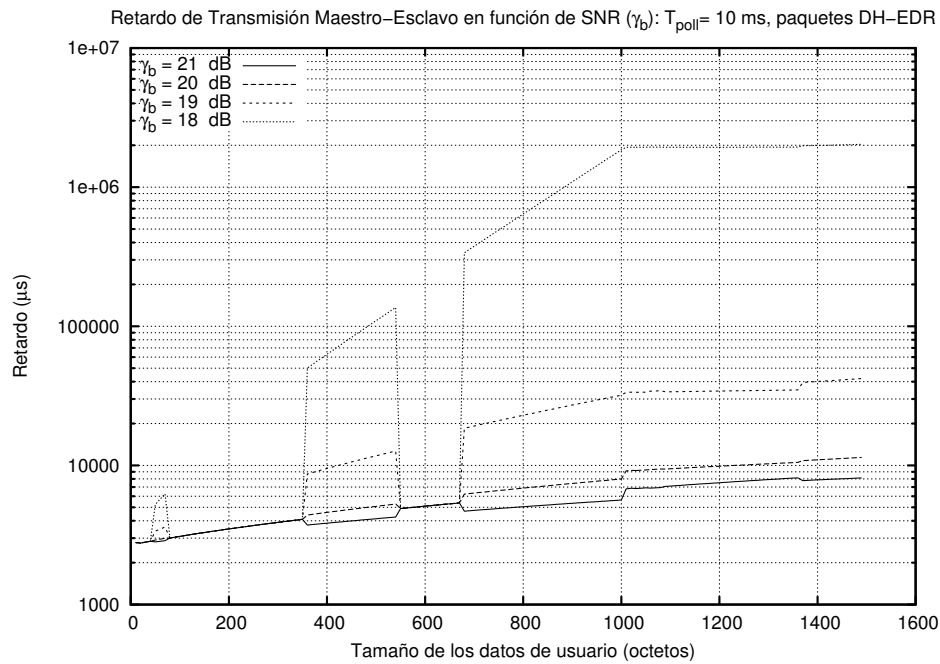


Figura 4.18.: Efecto de la relación SNR en las transmisiones maestro-esclavo para el perfil SPP, utilizando la política de selección de paquetes EDR reflejada en la figura 4.5.

CAPÍTULO 5

Evaluación empírica del modelo teórico sin retransmisiones

5.1. Introducción

En el presente capítulo se exponen las medidas realizadas con dispositivos de distintos fabricantes para evaluar empíricamente el modelo teórico desarrollado suponiendo condiciones ideales, es decir, que no se producen pérdidas de paquetes que den lugar a retransmisiones (véase el apartado 4.3).

Se comienza, en la sección 5.2, describiendo el sistema de pruebas, especificando el sistema operativo y dispositivos utilizados, así como el procedimiento de medida. Después, en la sección 5.3, se prosigue exponiendo los resultados obtenidos con dispositivos del fabricante CSR (*Cambridge Silicon Radio*) [CSR], comparándolos con el retardo previsto teóricamente, y desarrollando la corrección del modelo necesaria para contemplar dos fenómenos que distorsionan el retardo medido: el efecto de la capa de transporte y del tamaño de los *buffers* del dispositivo. Por último, en las secciones 5.4 y 5.5 se presentan los resultados de las medidas realizadas con dispositivos de otros fabricantes: ISS (*Integrated System Solution Corp.*) y *Broadcom Corporation*.

5.2. Descripción del sistema de pruebas

5.2.1. Sistema operativo y dispositivos Bluetooth empleados

Las medidas del retardo que se exponen en el presente capítulo se han llevado a cabo con dispositivos Bluetooth, con interfaz USB, conectados a un único PC con Sistema Operativo Ubuntu. Inicialmente, se comenzó empleando la versión Ubuntu Linux 7.04 [Linux] y *kernel* de la versión 2.6.20-lowlatency. Posteriormente, con el objeto de realizar en paralelo la ejecución de pruebas, se preparó un segundo PC con una versión más actualizada: Ubuntu Linux 8.04 [Linux] y *kernel* de la versión 2.6.24-rt.

La razón de optar por Linux como sistema operativo es que integra una implementación de código abierto del *Bluetooth Host* de la arquitectura de protocolos Bluetooth (véase la figura 2.1): BlueZ [BlueZ] que, además de incluir los niveles HCI *Firmware* y L2CAP y, entre otros, el protocolo RFCOMM, proporciona:

- Un conjunto de utilidades de configuración de dispositivos Bluetooth.
- Una herramienta de monitorización de la comunicación entre el *Host* y el *Bluetooth Controller*: *hcidump*.
- Una interfaz de programación para lenguaje C, basada en *sockets*, que permite tanto el acceso directo a las capacidades del *Bluetooth Controller* a través de la interfaz HCI, como el establecimiento de conexiones a nivel de HCI, L2CAP o RFCOMM.

En relación con las versiones de Ubuntu Linux empleadas como plataforma del sistema de medidas, cabe mencionar que se distinguen por ofrecer una gran estabilidad en su comportamiento, característica que no se mantiene en versiones posteriores, como en las versiones Ubuntu 8.10 Intrepid Ibex o Ubuntu 9.10 Karmic Koala, lanzadas el 30 de octubre de 2008 y el 29 de octubre de 2009, respectivamente. Para ilustrar esta circunstancia, en el Anexo A se incluyen algunos de los resultados obtenidos bajo esta última versión, comparándolos con los de las versiones con las que se ha llevado a cabo la validación empírica.

Como dispositivos Bluetooth se han empleado módulos USB de distintos fabricantes: Ovislink, Hamma, Inves y Trust, que incorporan circuitos integrados del fabricante CSR de la serie BlueCore 4, conformes a la especificación Bluetooth 2.0 + EDR con dos versiones de *firmware* diferentes y, también, con dispositivos Bluetooth 2.1 + EDR. Los resultados de las medidas realizadas y su comparación con el modelo teórico desarrollado en el capítulo 4 se exponen en la sección 5.3. Adicionalmente, las medidas realizadas con dispositivos de otros fabricantes distintos a CSR, ISS y *Broadcom Corporation*, se presentan, respectivamente, en las secciones 5.4 y 5.5.

5.2.2. Procedimiento de medida

En las pruebas realizadas se ha seguido la misma metodología empleada en [Moron08], esto es, se ha utilizado una arquitectura cliente-servidor programada en lenguaje C sobre la pila de protocolos BlueZ, en la que el extremo emisor, maestro o esclavo, transmite datos periódicamente, mientras que el receptor calcula el tiempo transcurrido desde el inicio de la transmisión de los datos hasta el instante de recepción.

Con el objeto de garantizar la sincronización de los relojes del extremo origen y destino, las pruebas se han ejecutado conectando las interfaces Bluetooth a un mismo PC, como se muestra en la figura 5.1, en la que también puede observarse que los módulos Bluetooth se han introducido en una caja metálica a fin de evitar posibles interferencias. Al respecto cabe señalar que se ha constatado que los mecanismos de control de potencia de transmisión y de control automático de ganancia del receptor, soportados por los módulos Bluetooth, evitan problemas en las transmisiones atribuibles a las reflexiones internas provocadas por la caja metálica, no viéndose, por tanto, alterado el resultado de las medidas.

Concretamente, el procedimiento de medida ha consistido en ejecutar, por cada prueba, dos programas (escritos en lenguaje C) sobre el PC: un cliente, que actúa como maestro de la *piconet*, y un servidor, que desempeña el papel de esclavo. Al programa cliente se le especifican, como argumentos, los siguientes parámetros:

- Modo de funcionamiento: Transmisión o Recepción.

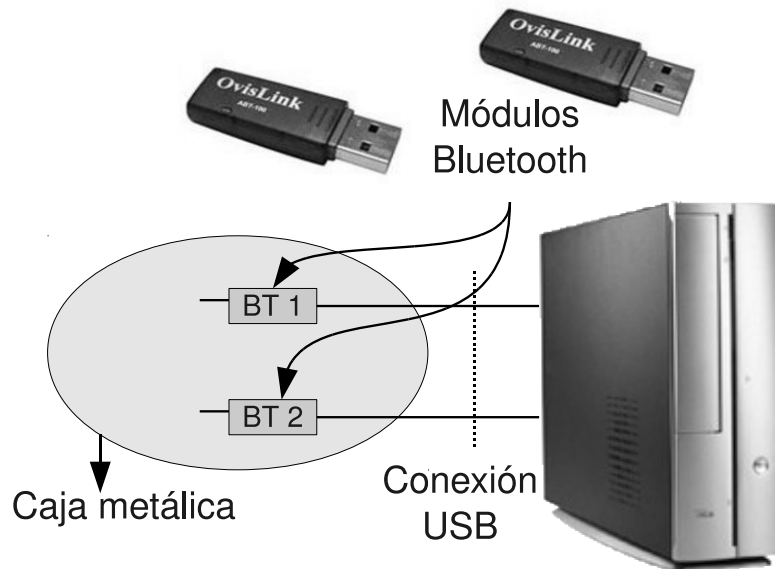


Figura 5.1.: Esquema general de pruebas empleado para la caracterización del retardo de transmisión Bluetooth, utilizado para evitar las interferencias

- Longitudes mínima y máxima de los datos a transmitir, así como el incremento entre longitudes consecutivas.
- Número de repeticiones para cada transmisión.
- Intervalo, en μs , entre transmisiones de datos consecutivas.
- Valor del parámetro *Latency* (equivalente a T_{poll}) a utilizar en la configuración de calidad de servicio.
- Paquetes seleccionados para la transmisión de datos: DM, DH en el modo básico o DH en el modo EDR.

Tras establecerse la conexión RFCOMM entre cliente y servidor, el cliente transmite al servidor la configuración (los parámetros de la prueba), de forma que una vez confirmada, el transmisor (cliente o servidor, según el modo de funcionamiento) realiza, a intervalos fijos, tantas transmisiones de datos de usuario como indique el parámetro «Número de

Tabla 5.1.: Valores de los parámetros comunes a todas la medidas

Parámetro	Valor
Longitud mínima	10 octetos
Longitud máxima	1500 octetos
Incremento entre longitudes	10 octetos
Nº. de repeticiones	1000
Intervalo entre transmisiones consecutivas	100000 μ s

repeticiones» por cada uno de los tamaños comprendidos entre las longitudes mínima y máxima especificadas, aumentando la longitud entre medidas según indica el parámetro «Incremento entre longitudes». Para los parámetros comunes a todas las medidas, por regla general, se han empleado los valores indicados en la tabla 5.1. Sólo en determinados casos en los que se ha querido validar algún ajuste del modelo se ha modificado alguno de los parámetros.

Para el cálculo del retardo de transmisión, como parte de los datos transmitidos, se incluye la marca temporal del instante de generación, de modo que el extremo de recepción, en un fichero creado para cada longitud de datos de usuario, almacena en cada repetición la diferencia entre el instante de recepción y el de transmisión. El resto de la información de usuario transmitida, hasta completar la longitud deseada en la medida, son octetos de relleno de valor constante.

Una vez concluida la medida, cada uno de los ficheros generados es procesado para obtener el valor mínimo, el máximo, la media y la desviación estándar de las series correspondientes a los retardos obtenidos en las 1000 iteraciones. Por último, cabe mencionar que el procesamiento de los resultados se ha llevado a cabo con una serie de programas (*scripts*) escritos en *Python*, que hacen uso de la biblioteca estadística *Python-Stats* [Strangman00].

5.3. Dispositivos CSR

Es conveniente aclarar que el grueso de las medidas presentadas en esta sección se han llevado a cabo con dispositivos Bluetooth 2.0 + EDR debido a que en el momento de su ejecución los dispositivos Bluetooth 2.1 + EDR no estaban disponibles comercialmente. No obstante, a posteriori, se han realizado medidas con dispositivos Bluetooth 2.1 + EDR con el objeto de probar la validez de los resultados, comparándolos con los obtenidos con dispositivos Bluetooth 2.0 + EDR para poner de manifiesto la ausencia de diferencias.

A continuación, en el apartado 5.3.1, se comienza comparando las medidas realizadas con las dos versiones de dispositivos del fabricante CSR Bluetooth 2.0 + EDR disponibles comercialmente, entre sí, y con el modelo teórico que, en una primera aproximación, no responde a la realidad, dado que aparecen efectos no contemplados en el modelo teórico. Del análisis de los resultados, como se explica en los siguientes epígrafes, cabe resaltar que, en el sentido de transmisión maestro-esclavo, para una de las versiones de *firmware*, la más extendida comercialmente, existe una dependencia del retardo medido respecto del valor del parámetro ACL MTU de módulo Bluetooth, parámetro que representa el tamaño de los *buffers* ACL en el que se almacenan los datos para su transmisión.

Precisamente, el tamaño de los *buffers* HCI, junto con el número de estos, son los parámetros que se especifican en el estándar Bluetooth en relación con el control de flujo entre el *Host* y el módulo Bluetooth. En concreto, se define un comando HCI (*HCI_Read_Buffer_Size*) para obtener ambos valores, tanto para enlaces lógicos ACL como para SCO [Bluetooth SIG04, parte H:1].

Es conveniente tener presente que, dado que el tamaño del *buffer* HCI limita el bloque de información máximo que puede transferirse desde el *Host* hasta el dispositivo Bluetooth, puede asimilarse a la unidad máxima de transmisión asociada al tipo de enlace que corresponda (ACL o SCO). Así, para los enlaces ACL (objeto de este estudio), en lo sucesivo se va a denotar como ACL MTU.

En un dispositivo comercial, en general, estos valores (ACL MTU y número de *buffers*) no pueden ser alterados por el usuario, ya que el fabricante los ajusta, en principio, para obtener el mejor rendimiento, teniendo en cuenta la memoria disponible en el módulo

Tabla 5.2.: Valores configurados del parámetro ACL MTU, tamaño de cada *buffer* del dispositivo Bluetooth, junto con el número de elementos de *buffer*.

ACL MTU (tamaño de <i>buffer</i>)	nº <i>buffers</i>
200	12
256	12
310	10
350	8
384	8
420	6
512	6
680	4
740	4
1021	3

Bluetooth. De hecho, en la especificación HCI no se contempla ningún comando para su modificación. No obstante, para dispositivos del fabricante CSR, BlueZ integra la herramienta *bccmd* que permite configurar un valor distinto al que traen por defecto, tanto del número de *buffers* como del tamaño de cada uno de ellos (ACL MTU), de forma que su producto no puede exceder el tamaño de memoria disponible en el dispositivo, cuyo mínimo se estima en unos 3100 octetos. Concretamente, en la tabla 5.2 se especifican, para cada posible valor del parámetro ACL MTU utilizado en las pruebas, el número de posiciones de dicho tamaño configuradas.

Adicionalmente, en el apartado 5.3.2, se evalúa el efecto de la configuración de la calidad de servicio, también, para dos versiones de *firmware*. En el apartado 5.3.3, se describe cómo se ha caracterizado la capa de transporte para, seguidamente, en el apartado 5.3.4 corregir el modelo teórico presentado en el capítulo 4, a fin de considerar el efecto de la capa de transporte en el cálculo del retardo, tanto si se utilizan paquetes DH en el modo EDR, como DM y DH en el modo básico. Este ajuste del modelo se realiza anulando la dependencia del parámetro ACL MTU, configurándolo a un valor igual o superior a la máxima capacidad de un paquete de 5 *slots*.

Con posterioridad, en el apartado 5.3.5, se refina el modelo para contemplar la dependencia del parámetro ACL MTU detectada en el sentido maestro-esclavo.

En última instancia, en el apartado 5.3.6 se explican los motivos por los que las correcciones realizadas, tanto para el modo básico como para el modo EDR, se ajustan a la realidad para longitudes de datos inferiores a los 1008 octetos y sobreestiman el retardo a partir de este límite. A la vez que se analiza este resultado, se propone cómo lograr una mejor aproximación.

Para finalizar, en el apartado 5.3.7, se recopilan todas las modificaciones realizadas en el modelo, al tiempo que se repasan las principales conclusiones extraídas sobre el comportamiento de los dispositivos CSR.

5.3.1. Evaluación inicial: Análisis del efecto del parámetro ACL MTU

Los resultados de las medidas presentadas a continuación se han realizado con dispositivos del fabricante CSR (*Cambridge Silicon Radio*) [CSR] de la serie *BlueCore 4*, conformes a la especificación Bluetooth 2.0 + EDR y con las siguientes versiones y revisiones del *firmware* HCI y del nivel LM:

- *Versión de HCI: 2.0 (0x3); Revisión de HCI: 0xc5c*
- *Versión del nivel LM: 2.0 (0x3); Subversión del nivel LM: 0xc5c*

En el sentido de transmisión maestro-esclavo, el primer resultado arrojado por las medidas es que, en una primera aproximación, esta versión de *firmware* presenta una dependencia respecto del valor del parámetro ACL MTU configurado. Este efecto se ilustra en la figura 5.2, para un valor de T_{poll} de 25 ms, lo que equivale a no haber configurado calidad de servicio, y para los siguientes valores del parámetro ACL MTU: 1021, 512, 384 y 310 octetos. Es preciso aclarar que los valores 310 y 384 son los configurados por defecto en los dispositivos comerciales y que los valores 512 y 1021 han sido establecidos mediante la herramienta de configuración para dispositivos CSR *bccmd*, integrada en BlueZ [BlueZ].

Nótese que el menor retardo medido se obtiene cuando el parámetro ACL MTU se configura a un valor de 512 octetos. Adicionalmente, tal y como puede observarse en cada

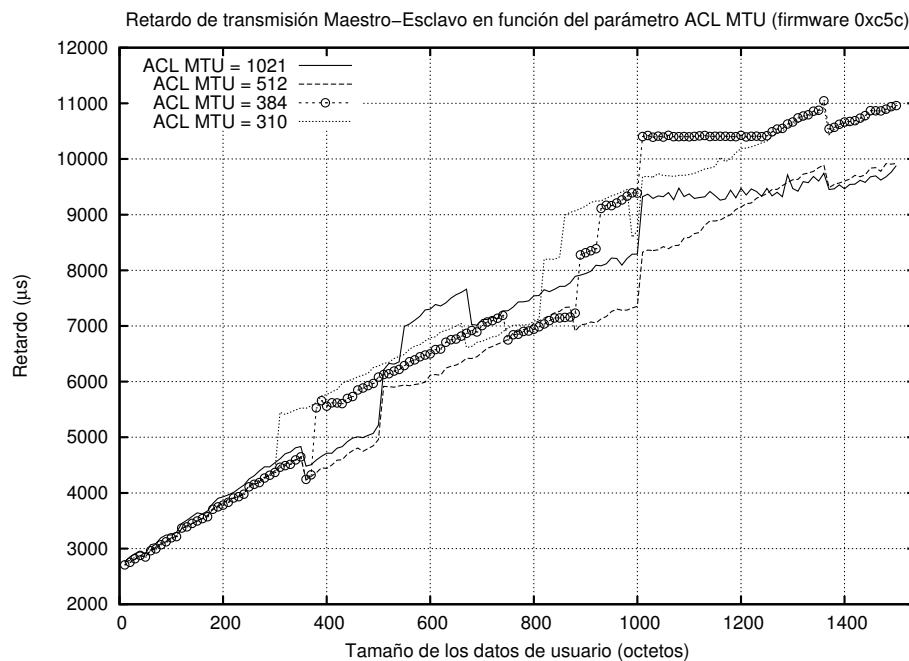


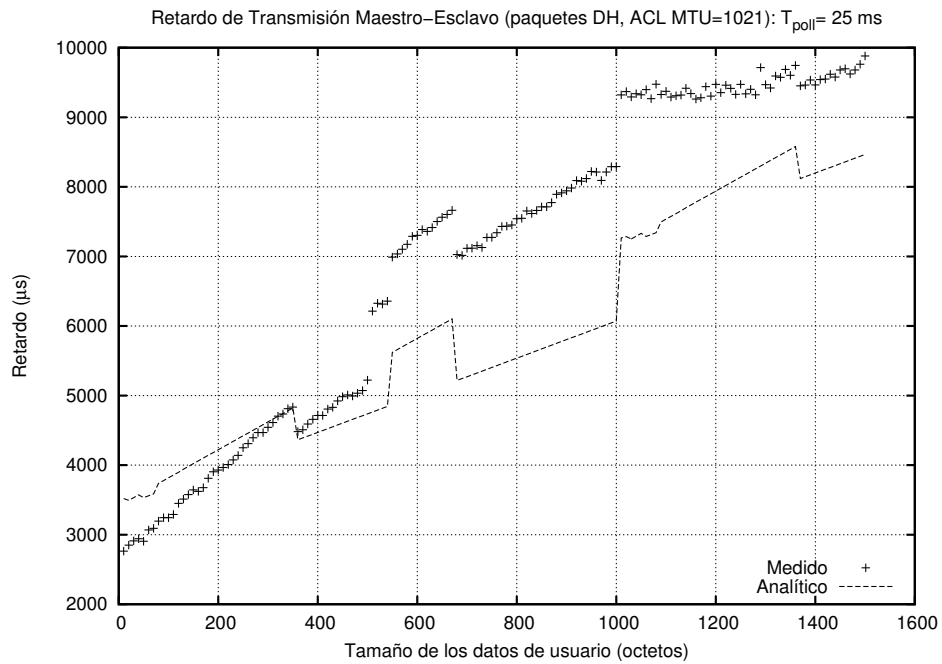
Figura 5.2.: Efecto de la configuración del parámetro ACL MTU para la versión del *firmware* 0xc5c

una de las gráficas de las figuras 5.3 y 5.4, que comparan los retardos previstos teóricamente y los medidos para cada uno de los valores del parámetro ACL MTU configurado (1021, 512, 384 y 310 octetos), cabe destacar que no existe coincidencia en ninguno de los casos.

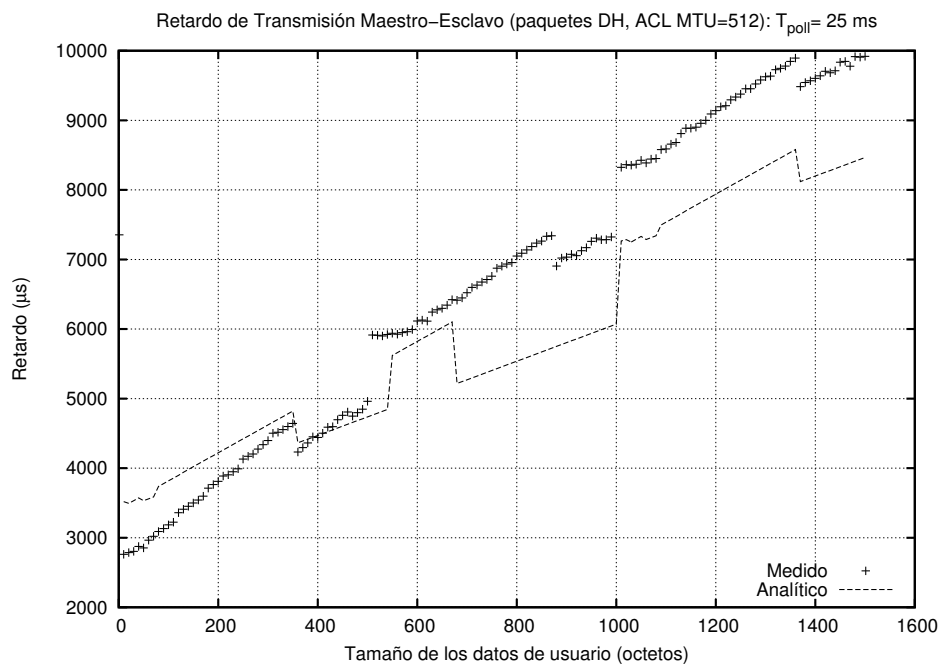
Antes de proseguir con el análisis de los resultados presentados, cabe destacar que este resultado contrasta con el obtenido, también en el sentido de transmisión maestro-esclavo, para una versión anterior del *firmware* de dispositivos del fabricante CSR, también pertenecientes a la serie *BlueCore 4*:

- Versión de HCI: 2.0 (0x3); Revisión de HCI: 0x7a6
- Versión del nivel LM: 2.0 (0x3); Subversión del nivel LM: 0x7a6

El hecho es que esta versión, 0x7a6, a diferencia de la 0xc5c, no presenta dependencia respecto del valor del parámetro ACL MTU. A modo de ejemplo, baste comparar los retardos de ambas versiones de *firmware*, sin tener en cuenta el efecto de la calidad de servicio ($T_{poll} = 25 \text{ ms}$), presentados en la figura 5.5 para un valor del parámetro ACL MTU de 384 octetos.

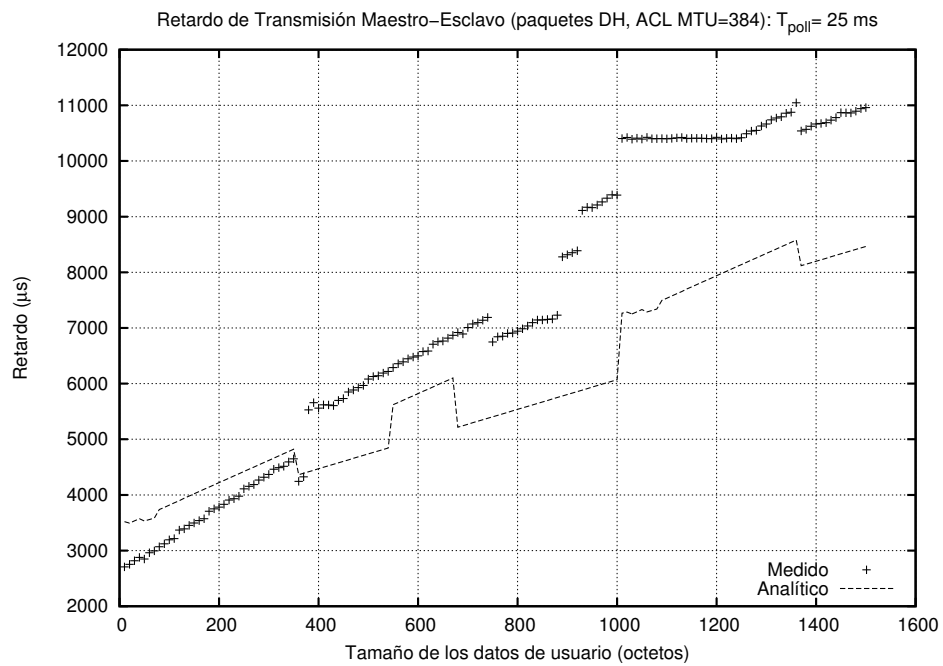


(a) ACL MTU de 1021 octetos

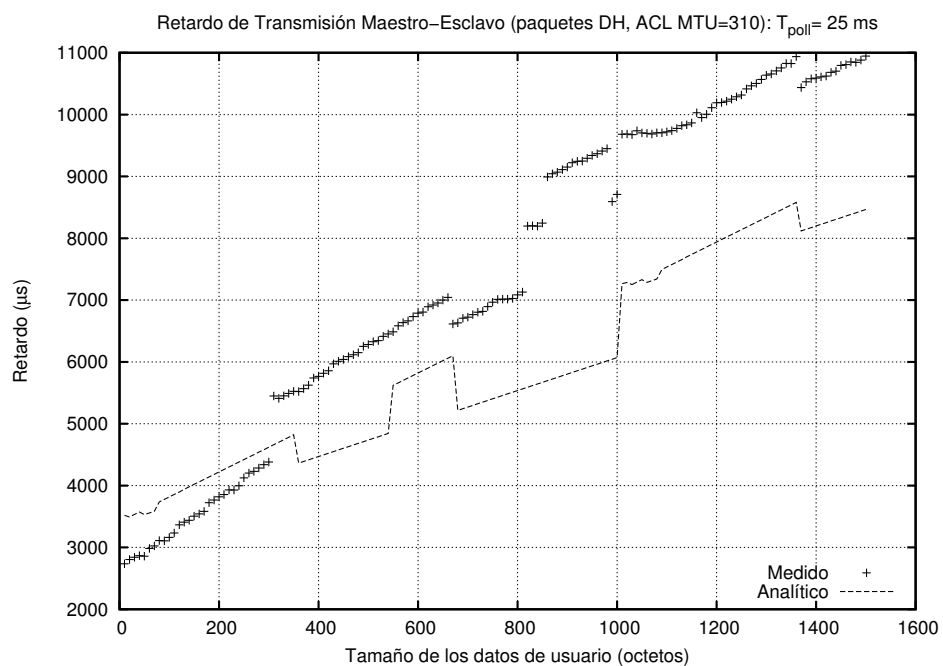


(b) ACL MTU de 512 octetos

Figura 5.3.: Comparación del retardo medio de transmisión maestro-esclavo previsto teóricamente con el medido para un valor del parámetro ACL MTU de 1021 y de 512 octetos (Revisión de HCI: 0xc5c).



(a) ACL MTU de 384 octetos



(b) ACL MTU de 310 octetos

Figura 5.4.: Comparación del retardo medio de transmisión maestro-esclavo previsto teóricamente con el medido para un valor del parámetro ACL MTU de 384 y de 310 octetos (Revisión de HCI: 0xc5c).

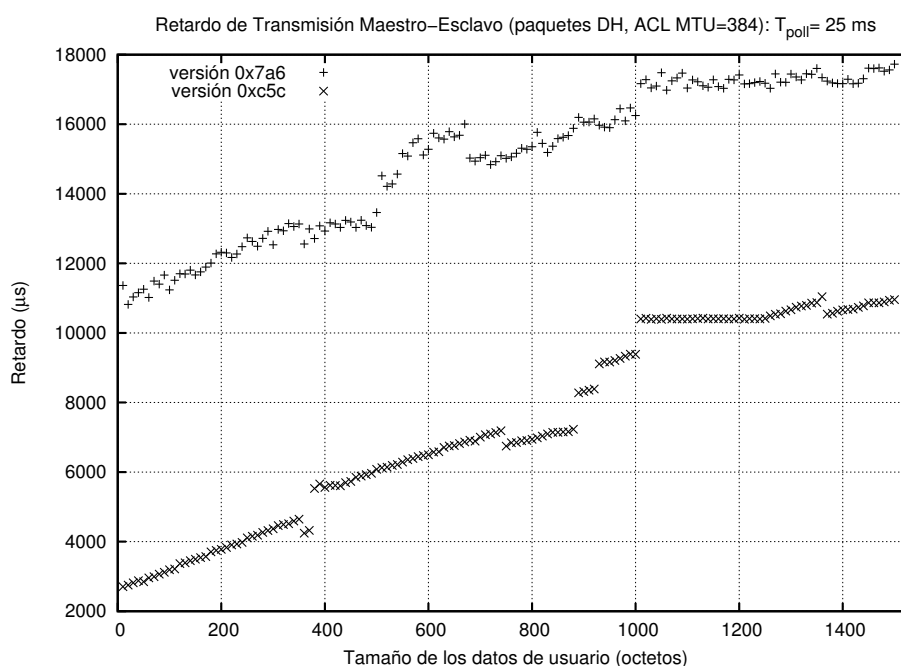


Figura 5.5.: Comparación de los retardos medios de transmisión obtenidos para las versiones de *firmware* 0x7a6 y 0xc5c, con un valor del parámetro ACL MTU de 384 octetos, en ambos casos.

Puede observarse que difieren completamente, ya que no coinciden ni las pendientes de las curvas de retardo, ni tampoco los saltos verticales (incrementos y decrementos de retardo) asociados a los cambios de paquetes y de velocidades de modulación.

En cualquier caso, tal y como se ilustra en las gráficas de la figura 5.6, el retardo previsto teóricamente tampoco se ajusta al medido para la versión de *firmware* 0x7a6, coincidiendo en todos los casos, es decir, para todos los posibles valores del parámetro ACL MTU, con la medida obtenida para la versión de *firmware* 0xc5c en el caso particular de la ACL MTU de 1021 octetos, tal y como se ilustra, a modo de ejemplo, en la figura 5.7. En esta figura, en la que se representa el retardo medio y su desviación típica para cada longitud, puede apreciarse que, además del *offset* en el que se diferencian, la medida obtenida para la versión de *firmware* 0x7a6 presenta una mayor dispersión, siendo en este caso la desviación típica del orden de $4000\ \mu\text{s}$, que es 5,71 veces superior a la correspondiente a la versión de *firmware* 0xc5c, $700\ \mu\text{s}$.

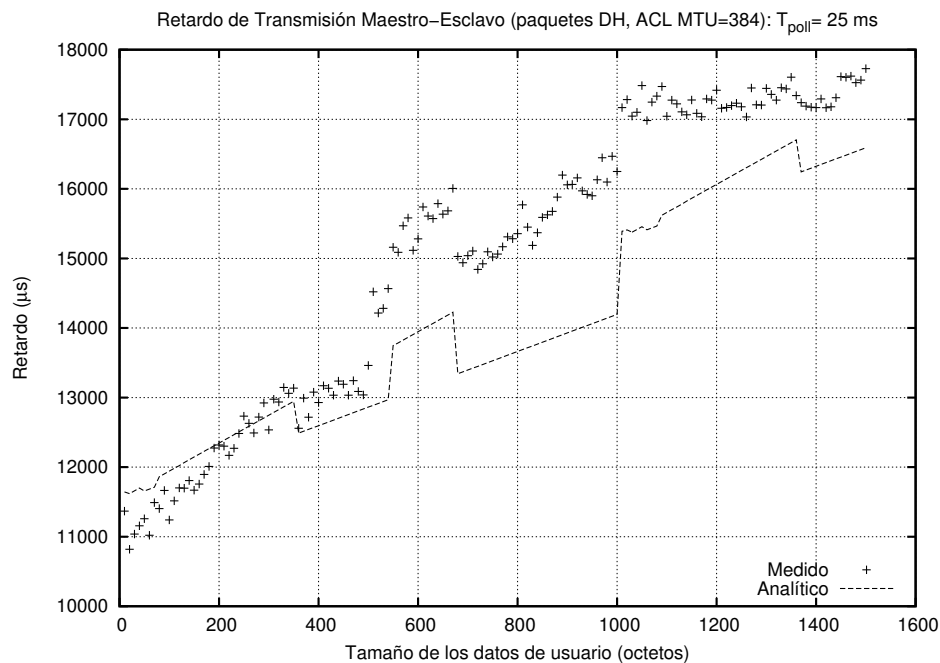


Figura 5.6.: Comparación del retardo medio de transmisión maestro-esclavo previsto teóricamente con el retardo medido para la versión de *firmware* 0x7a6, con un valor del parámetro ACL MTU de 384 octetos.

Por contra, cabe destacar que se ha corroborado que la dependencia del parámetro ACL MTU detectada en el sentido maestro-esclavo con la versión de *firmware* 0xc5c, ausente en la versión previa 0x7a6, persiste en versiones posteriores. Concretamente, se ha repetido la medida con dispositivos del fabricante CSR Bluetooth 2.1 + EDR, con la versión de *firmware* siguiente:

- Versión de HCI: 2.1 (0x4); Revisión de HCI: 0x12e7
- Versión del nivel LM: 2.1 (0x4); Subversión del nivel LM: 0x12e7

Los resultados obtenidos, tanto con los dispositivos Bluetooth 2.0 + EDR como con Bluetooth 2.1 + EDR, coinciden en gran medida, hasta el punto de que son indistinguibles, según puede comprobarse en las figuras 5.8, 5.9a y 5.9b, para un valor del parámetro ACL MTU de 1021, 512 y de 384 octetos, respectivamente.

Hasta este punto, todos los resultados expuestos son los obtenidos configurando el uso de paquetes DH del modo básico y EDR, es decir, excluyendo los paquetes DM. Si, por el

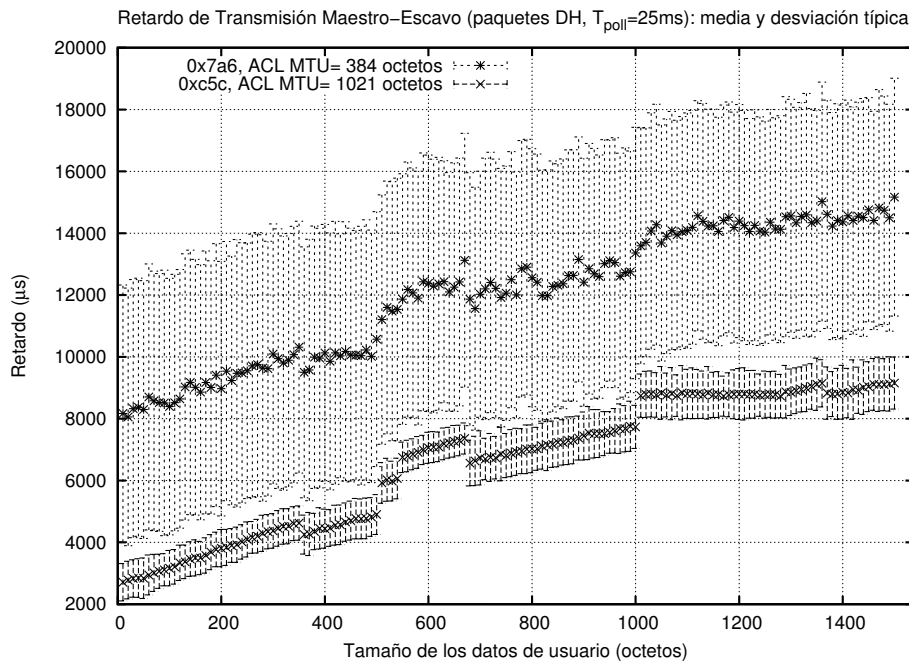


Figura 5.7.: Comparación de los retardos medios de transmisión medidos y de su desviación típica para las versiones de *firmware* 0x7a6 y 0xc5c, con valores del parámetro ACL MTU de 384 octetos y 1021 octetos, respectivamente.

contrario, en la programación de las medidas se deshabilitan los paquetes DH y se permiten los paquetes DM, puede comprobarse que, en este caso, para la versión de *firmware* 0xc5c, el valor del parámetro ACL MTU configurado también afecta al retardo cuando es inferior a la máxima capacidad de un paquete de 5 *slots*, DM5 (224 octetos). Esta dependencia se muestra en la figura 5.10, en la que se han representado los retardos medios medidos sin haber configurado la calidad de servicio y cuando el parámetro ACL MTU toma los valores de 200, 310, 512 y 680 octetos. Se aprecia que, para estos tres últimos valores, superiores a 224 octetos el retardo medido coincide, mientras que para el primer caso, que se encuentra por debajo, los incrementos abruptos del retardo (escalones) tienen lugar a los 200 octetos, antes de completarse un paquete DM5. De hecho, para los paquetes DM, en ausencia del efecto del parámetro ACL MTU, el modelo propuesto, a diferencia de lo que sucede con los paquetes DH en el modo EDR, se ajusta a la realidad incluso mejor que en el caso de dispositivos Bluetooth 1.1, según puede constatarse en la figura 5.11.

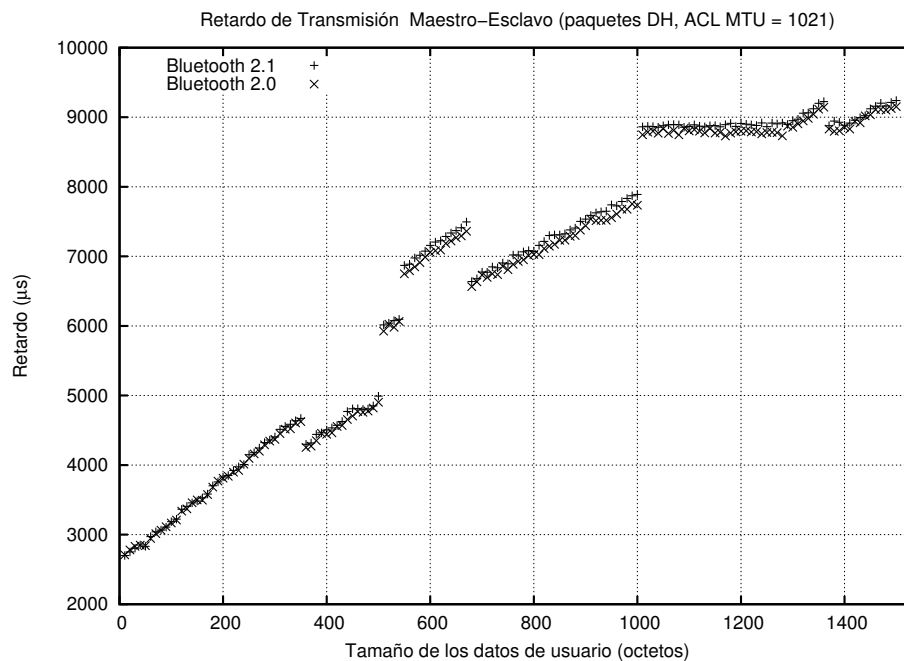
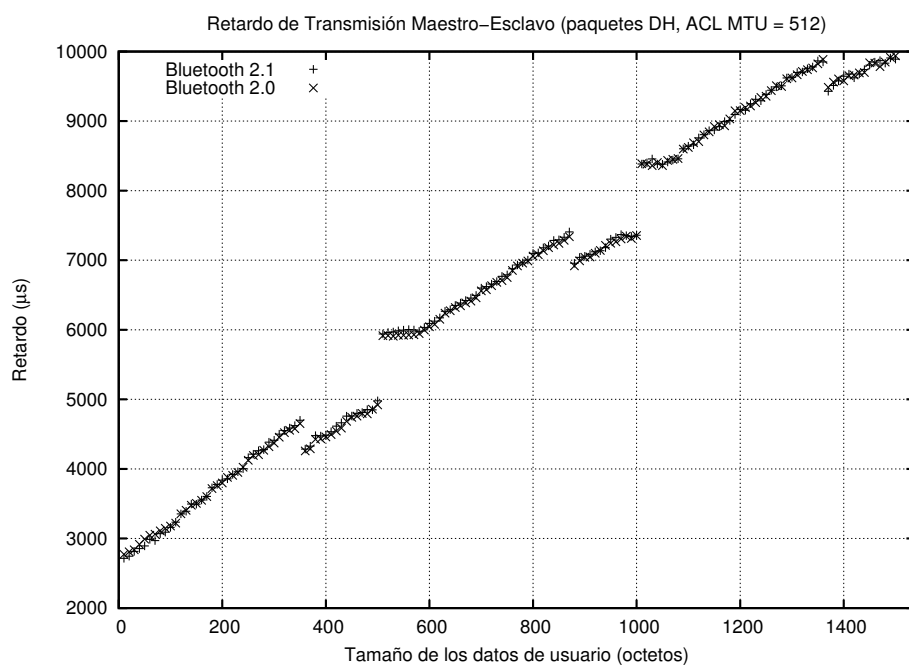


Figura 5.8.: Comparación de los retardos obtenidos en el sentido de transmisión maestro-esclavo con dispositivos Bluetooth 2.0 + EDR y Bluetooth 2.1 + EDR para un valor del parámetro ACL MTU de 1021 octetos.

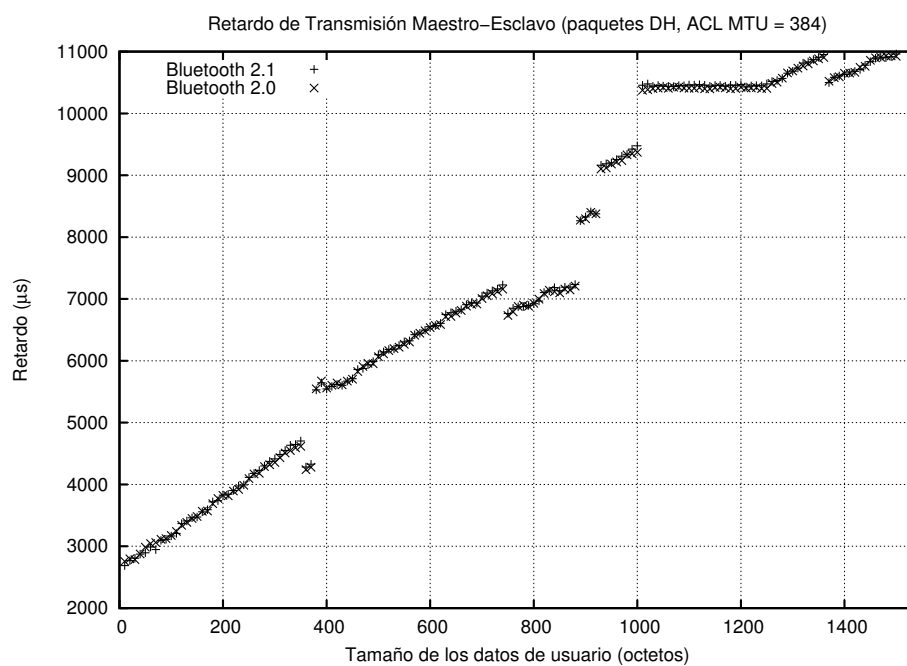
Además, en la figura 5.11 puede apreciarse que la medida realizada con dispositivos Bluetooth 1.1 presenta un comportamiento escalonado inexistente en la llevada a cabo con dispositivos Bluetooth 2.0 + EDR, tanto si la versión de *firmware* es la 0xc5c, como si se trata de la 0x7a6 (véase la figura 5.12).

De la comparación de estos resultados con los obtenidos con dispositivos Bluetooth 1.1, y de las medidas realizadas en el modo EDR, parece desprenderse la hipótesis de que el comportamiento escalonado en la versión Bluetooth 1.1 está relacionado con el hecho de que el parámetro ACL MTU de los dispositivos utilizados es de 192 octetos, valor inferior a la capacidad de un paquete DM5: 224 octetos. Como aval de este argumento, cabe destacar que los valores del parámetro en cuestión, que vienen configurados en los dispositivos Bluetooth 2.0 + EDR por defecto, son superiores a este límite (224 octetos).

Para ratificar esta hipótesis se ha repetido la medida de retardo permitiendo exclusivamente el uso de paquetes DH del modo básico. Los resultados, con tan solo un análisis preliminar, confirman que el efecto del parámetro ACL MTU se pone de



(a) ACL MTU de 512 octetos



(b) ACL MTU de 384 octetos

Figura 5.9.: Comparación de los retardos obtenidos en el sentido de transmisión maestro-esclavo con dispositivos Bluetooth 2.0 + EDR y Bluetooth 2.1 + EDR.

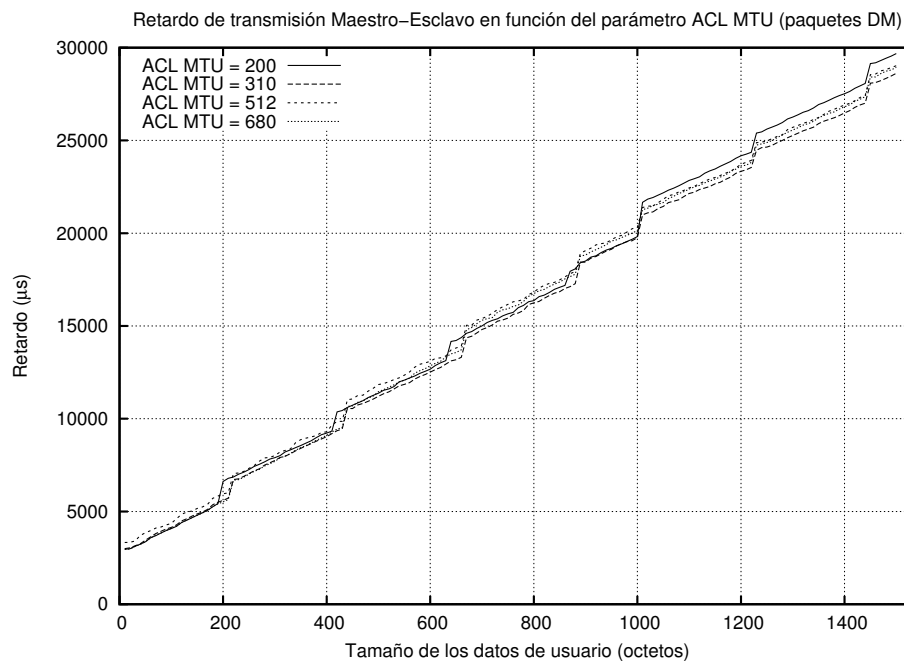
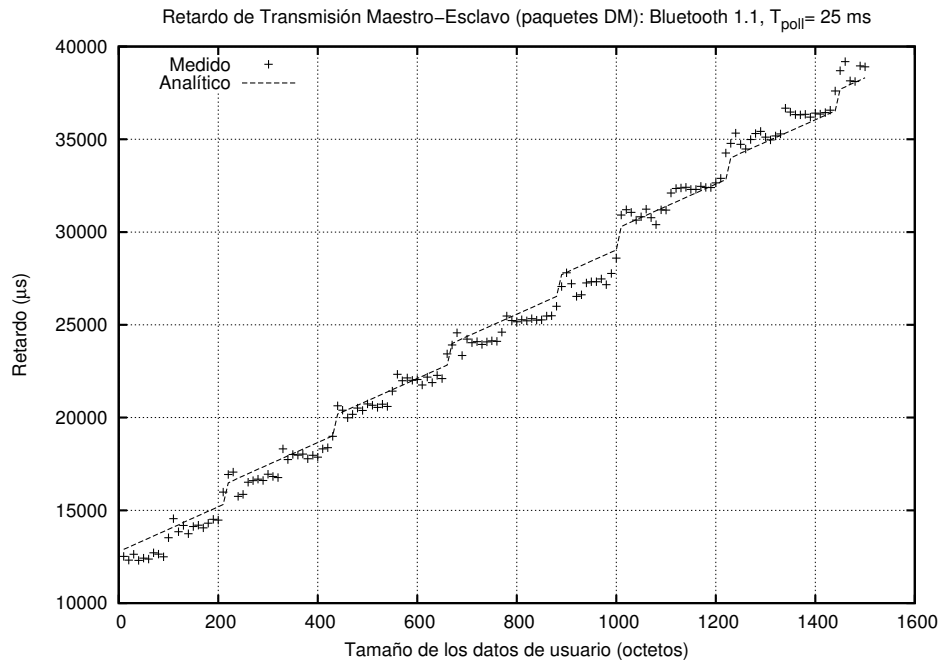


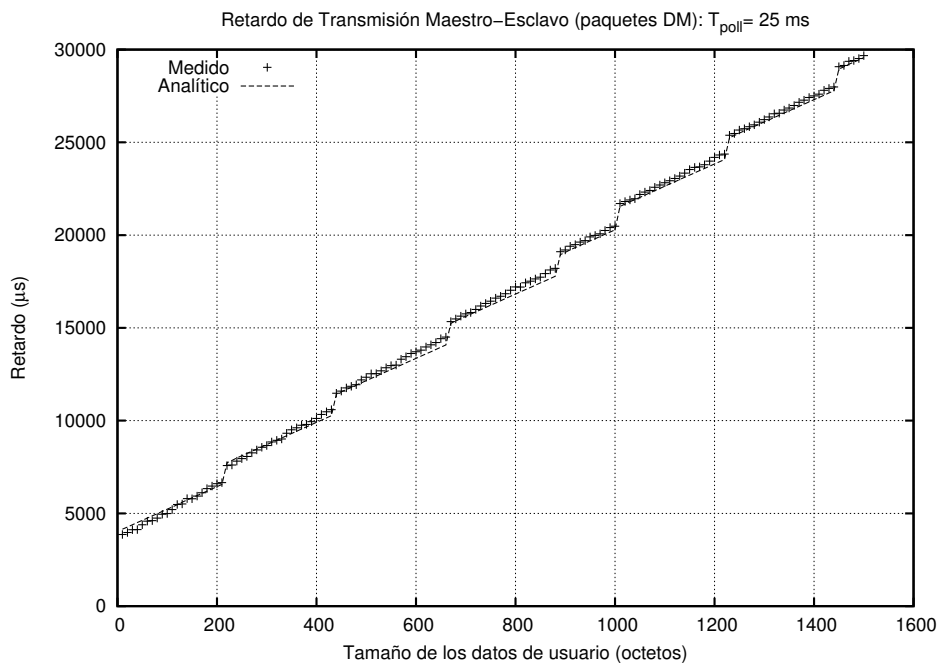
Figura 5.10.: Comparación de los retardos medios de transmisión obtenidos para la versión de *firmware* 0xc5c, utilizando paquetes DM, cuando el parámetro ACL MTU toma los valores de 200, 310, 512 y 680 octetos

manifiesto cuando se configura a un valor inferior al máximo número de octetos que pueden transportarse en un paquete de 5 *slots*, DH5 (339 octetos). Esta dependencia se muestra en las gráficas de la figura 5.13, correspondientes al retardo medido cuando el parámetro ACL MTU toma los valores de 256 y 310 octetos, inferiores a 339 octetos, y de 384, 512 y 680 octetos, que se encuentran por encima de la máxima capacidad de un paquete DH5. Puede observarse que, al igual que con los paquetes DM, para los valores inferiores al tamaño máximo de un paquete DH5 los incrementos abruptos del retardo se producen a la longitud que coincide con el límite impuesto por el parámetro ACL MTU.

En lo que respecta al sentido esclavo-maestro, con cualquiera de las versiones de *firmware* —0x7a6 y 0xc5c— el comportamiento obtenido es equivalente al que presentan en el sentido maestro-esclavo la versión de *firmware* 0x7a6, tal y como puede comprobarse en la figura 5.14. Cada una de las gráficas incluidas en esta figura corresponden a los retardos medios medidos para un valor de T_{poll} de 25 ms y para los siguientes valores del parámetro ACL MTU: 1021, 512, 384 y 310 octetos. Como puede constatarse a partir de las medidas



(a) Dispositivos Bluetooth 1.1



(b) Dispositivos Bluetooth 2.0 + EDR: revisión de HCI 0xc5c y ACL MTU de 512 octetos

Figura 5.11.: Comparación del retardo medio previsto teóricamente y del retardo medio de transmisión obtenido con dispositivos Bluetooth 1.1 y Bluetooth 2.0 + EDR (Revisión de HCI 0xc5c)

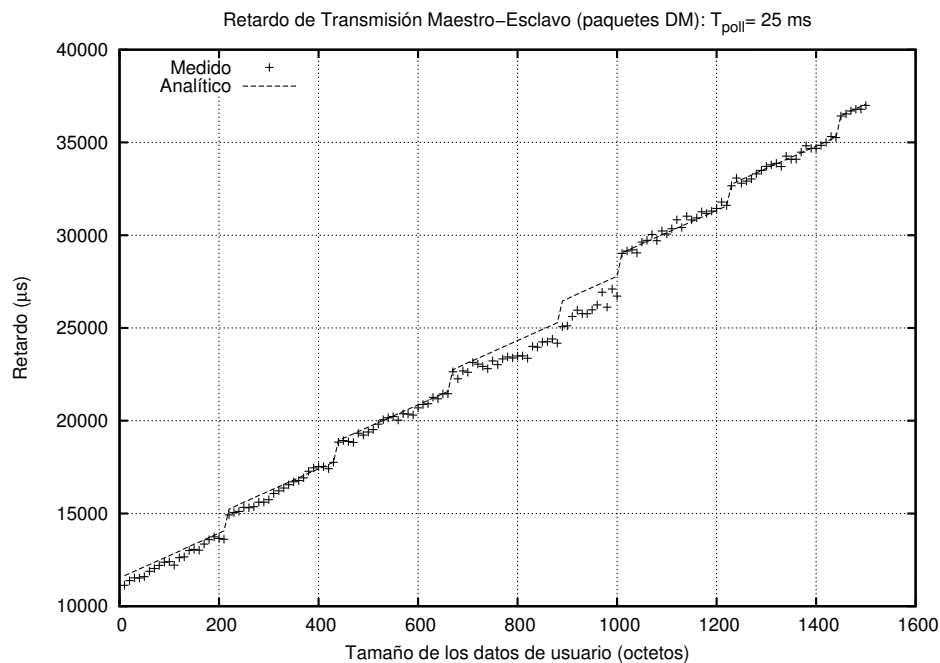


Figura 5.12.: Comparación del retardo medio previsto teóricamente y del retardo medio de transmisión obtenido con dispositivos Bluetooth 2.0 + EDR (Revisión de HCI 0x7a6)

realizadas, en el sentido de transmisión esclavo-maestro no existe una dependencia del valor del parámetro ACL MTU configurado.

A modo de conclusión, cabe resaltar que al retardo de transmisión Bluetooth definido por el modelo propuesto se suman los siguientes efectos:

- El retardo asociado a la capa de transporte USB, que resulta ser independiente del sentido de transmisión, caracterizado según se describe en el apartado 5.3.3 e incorporado al modelo en el apartado 5.3.4.
- El retardo introducido por la fragmentación realizada de acuerdo con el valor del parámetro ACL MTU, efecto que se manifiesta tanto en el modo básico como en el modo EDR y que sólo está presente en el sentido de transmisión maestro-esclavo a partir de la versión de *firmware* 0xc5c (véase el apartado 5.3.5).

Antes de abordar el modelado de cada uno de estos factores, en el apartado 5.3.2, se analiza el efecto de la configuración de la calidad de servicio, para cada uno de los sentidos de transmisión y para las versiones de *firmware* 0xc5c y 0x7a6.

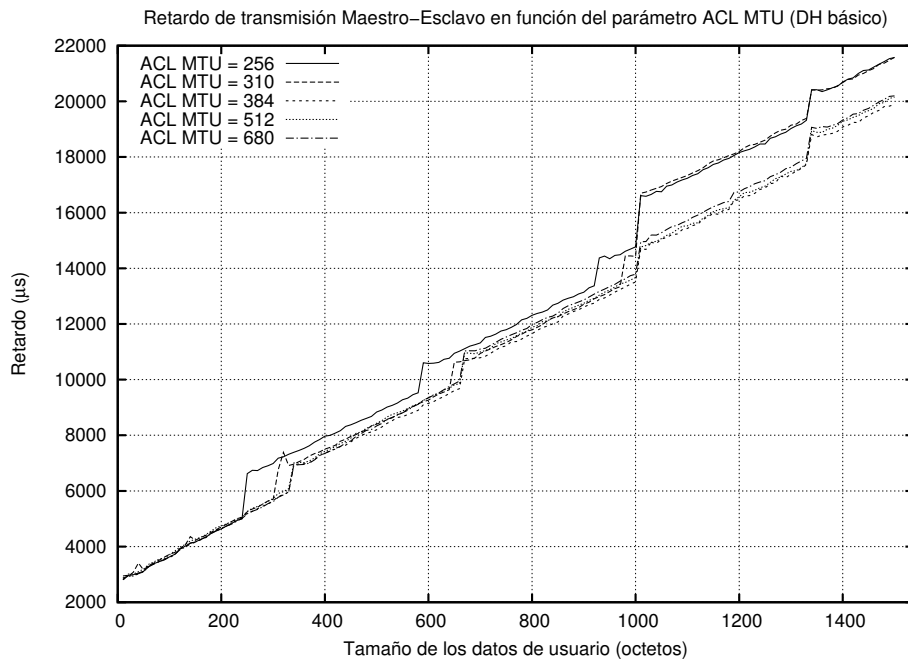


Figura 5.13.: Comparación de los retardos medios de transmisión obtenidos para la versión de *firmware* 0xc5c, utilizando paquetes DH del modo básico, cuando el parámetro ACL MTU toma los valores de 256, 310, 384, 512 y 680 octetos

5.3.2. Análisis del efecto de la configuración de servicio

Con el fin de configurar la calidad de servicio se ha empleado el comando *HCI_QoS_Setup* (véase el apartado 7.2.6 de la especificación [Bluetooth SIG04, vol. 2, part E]). Este comando, al igual que en la especificación Bluetooth 1.1 (apartado 4.6.6 de la especificación [Bluetooth SIG01, parte H:1]), permite especificar al nivel LM los parámetros de calidad de servicio relacionados con la especificación de flujo del nivel L2CAP, descritos en el apartado 2.2.1.4.

Tras el establecimiento de la conexión RFCOMM se ha utilizado el comando indicado, solicitando *Best Effort* como tipo de servicio y especificando, para el parámetro *Latency*, que es convertido de microsegundos a *slots* y empleado por el nivel LM directamente para intentar fijar a ese valor el periodo T_{poll} , los valores indicados en la tabla 5.3.

En los siguientes epígrafes se presentan los resultados obtenidos para cada sentido de transmisión y para las versiones de *firmware* 0x7a6 y 0xc5c, aunque es conveniente

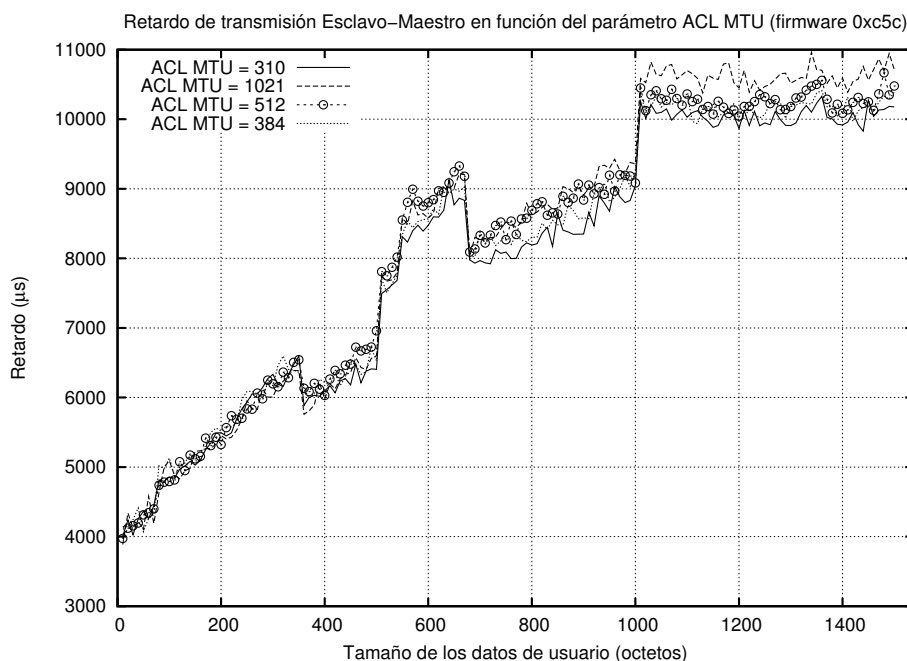


Figura 5.14.: Efecto de la configuración del parámetro ACL MTU para la versión del *firmware* 0xc5c en el sentido de transmisión esclavo-maestro

aclarar que el análisis de dichos resultados no se completa hasta el apartado 5.3.4.1, al estar supeditado a la inclusión del efecto de la capa de transporte, presentado en el apartado 5.3.3.

Tabla 5.3.: Valores del parámetro *Latency* (y del intervalo de *poll* al que equivale) para los que se ha evaluado el efecto de la configuración de la calidad de servicio

<i>Latency</i> (T_{poll})	Intervalo de <i>poll</i> (n_{poll})
5000 μs	8 slots
10000 μs	16 slots
15000 μs	24 slots
20000 μs	32 slots
25000 μs	40 slots

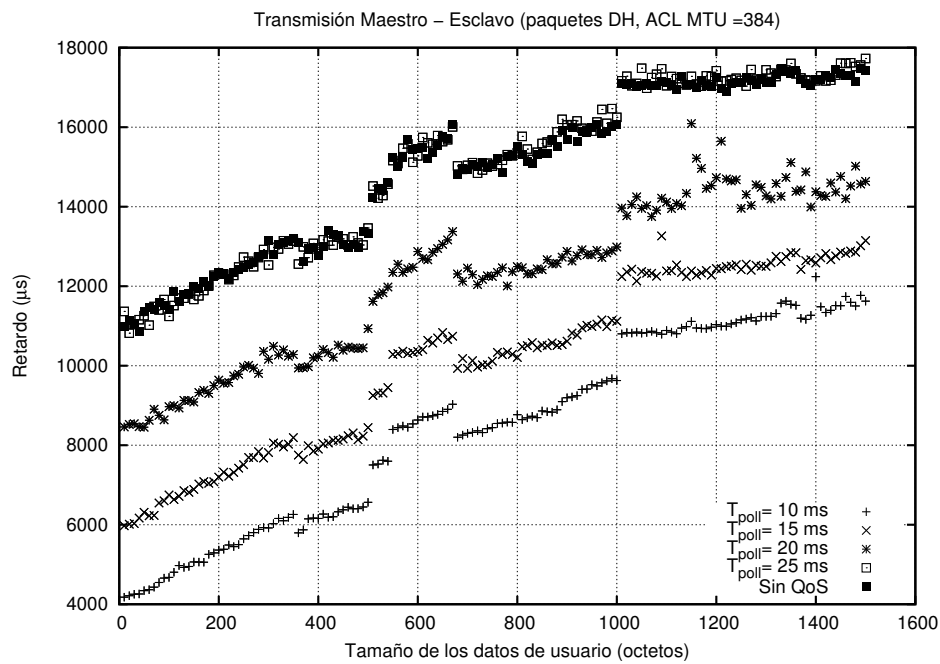
5.3.2.1. Sentido de transmisión maestro-esclavo

En el sentido maestro-esclavo, para la versión de *firmware* 0x7a6 y configurando el parámetro ACL MTU a un valor de 384 octetos, se han obtenido los resultados mostrados en la figura 5.15, tanto para paquetes DH como para paquetes DM y para todos los valores del parámetro *Latency* especificados en la tabla 5.3, a excepción del caso de 5 ms, valor para el que la medida no ha podido ser completada. Conviene aclarar que se ha seleccionado el caso de 384 octetos, a modo de ejemplo, ya que como se ha expuesto en el apartado 5.3.1, para la versión 0x7a6 no existe dependencia respecto del valor del parámetro ACL MTU empleado.

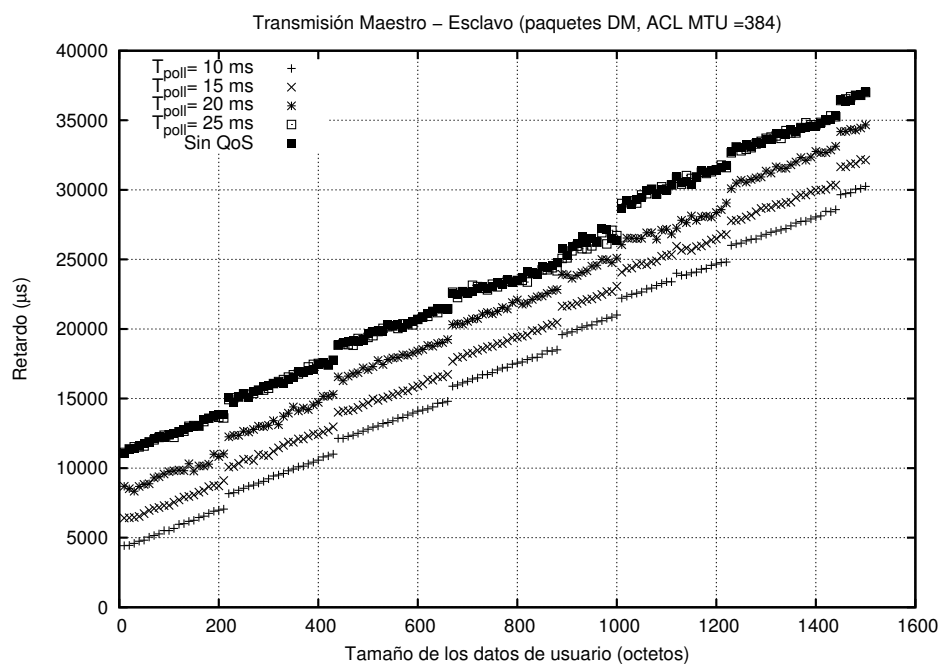
Puede comprobarse que, al igual que en Bluetooth 1.1 (véase el apartado 5.3.1.1 de [Moron08]), el efecto del valor del parámetro *Latency* configurado se traduce en un incremento constante del retardo de transmisión (*offset*), aunque a diferencia de lo que sucede para Bluetooth 1.1, el *offset* no se corresponde con $T_{poll}/2$, valor medio esperado por la acción del mecanismo de *poll*. Por ejemplo, para un valor de T_{poll} (asimilable al parámetro *Latency*) de 10 ms (16 *slots*), el *offset* obtenido para el retardo medio, en el caso de utilizar paquetes DH, es de 4 ms, inferior a $T_{poll}/2 = 5$ ms. Además, análogamente a Bluetooth 1.1, en la gráfica 5.15, puede apreciarse que no configurar la calidad de servicio es equivalente a establecer un intervalo de *poll* de 25 ms (T_{poll} de 25 ms).

En lo que concierne a la versión de *firmware* 0xc5c y para valores del parámetro ACL MTU de 1021, 512, 384 y 310 octetos se han obtenido, en el sentido maestro-esclavo, los resultados mostrados en las figuras 5.16a, 5.16b, 5.17a y 5.17b. Nótese que en todos los casos se muestran los resultados obtenidos sin haber configurado la calidad de servicio y habiéndola configurado para los valores del parámetro *Latency* especificados en la tabla 5.3, con la excepción del caso en el que dicho parámetro toma el valor de 5 ms, para el cual el retardo aumenta excesivamente, llegando al punto de interrumpirse la conexión Bluetooth. Puede observarse en las gráficas indicadas que el efecto de la configuración de calidad no se traduce en un incremento del retardo, llamando la atención el hecho de que las curvas presentan una variabilidad que causa que el retardo medio supere al obtenido cuando el valor del parámetro *Latency* se fija a 25 ms, correspondiendo a este caso, el menor retardo.

En cuanto a los resultados presentados en la figura 5.18 para los paquetes DM y la versión de *firmware* 0xc5c, cabe resaltar que no existe dependencia respecto del valor del

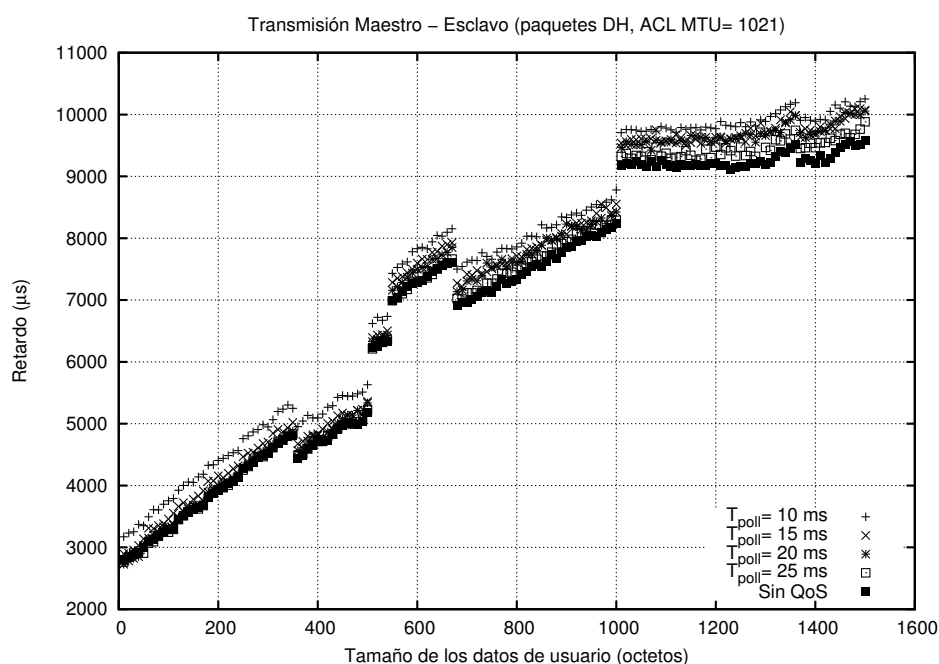


(a) Paquetes DH

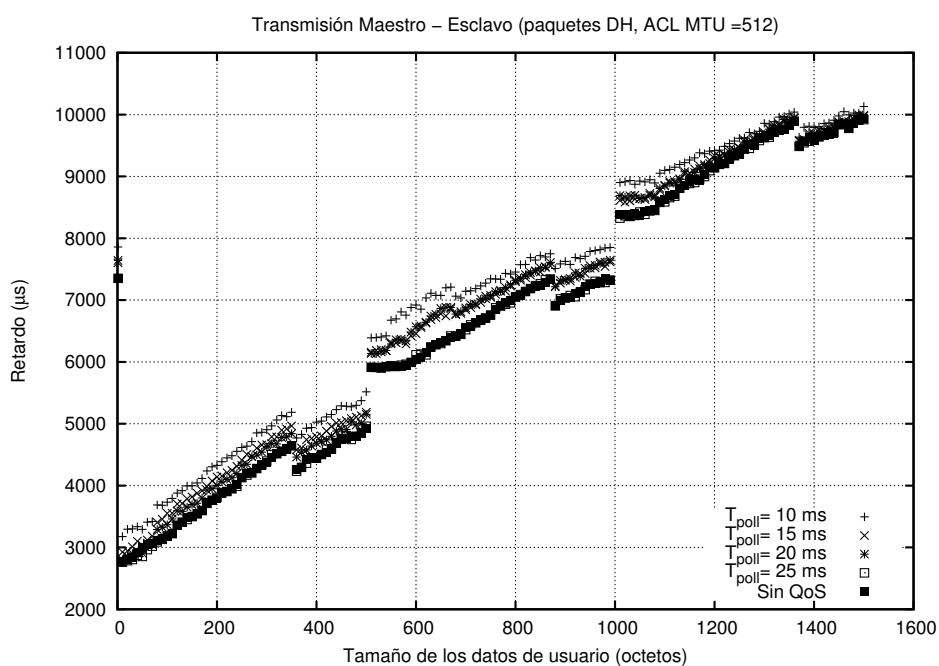


(b) Paquetes DM

Figura 5.15.: Efecto de la configuración de la calidad de servicio, en el sentido de transmisión maestro-esclavo, para la versión de *firmware* 0x7a6 y para un valor del parámetro ACL MTU de 384 octetos

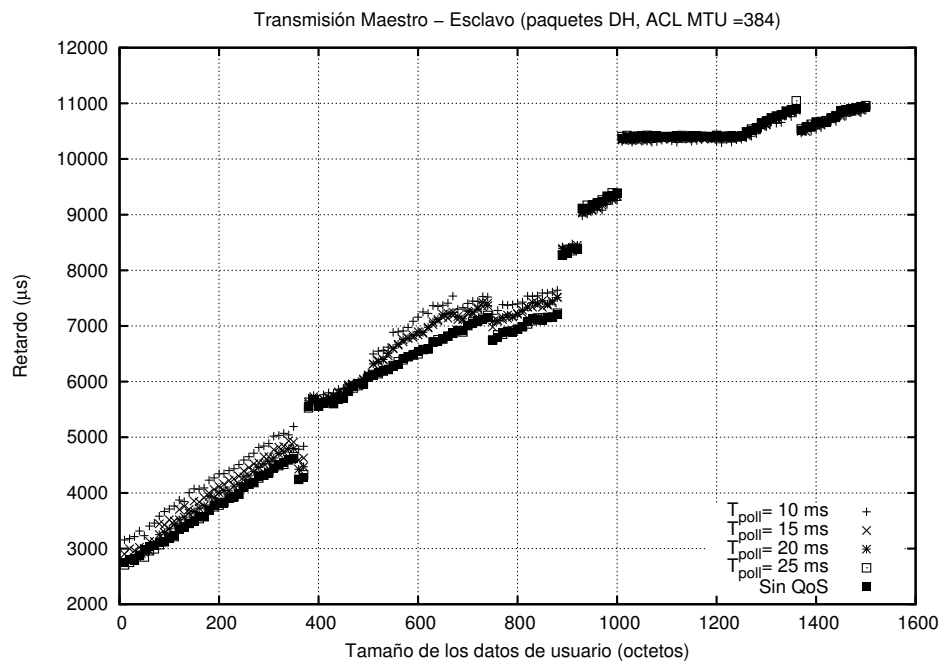


(a) ACL MTU de 1021 octetos

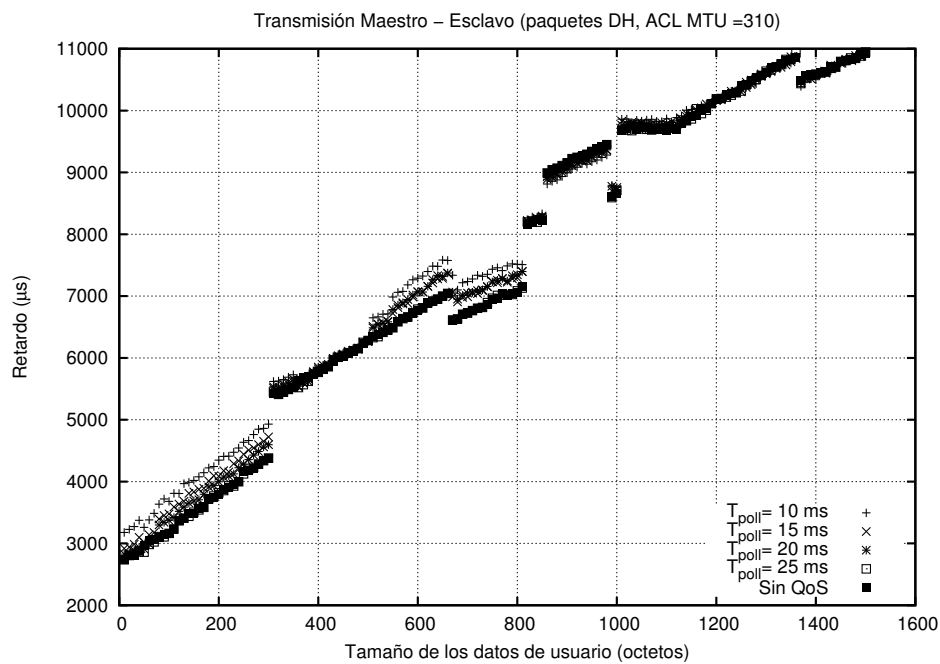


(b) ACL MTU de 512 octetos

Figura 5.16.: Efecto de la configuración de la calidad de servicio, en el sentido de transmisión maestro-esclavo, para la versión de *firmware* 0xc5c para paquetes DH y valores del parámetro ACL MTU de 1021 y 512 octetos.



(a) ACL MTU de 384 octetos



(b) ACL MTU de 310 octetos

Figura 5.17.: Efecto de la configuración de la calidad de servicio, en el sentido de transmisión maestro-esclavo, para la versión de *firmware* 0xc5c para paquetes DH y valores del parámetro ACL MTU de 384 y 310 octetos.

parámetro ACL MTU, ya que se ha configurado a 512 octetos, valor superior a la máxima capacidad de un paquete DM5. En la figura 5.18 también puede constatarse que, para esta versión, no se aprecia efecto de la configuración de calidad sobre el retardo medio, no existiendo *offset* dependiente del valor del parámetro *Latency*.

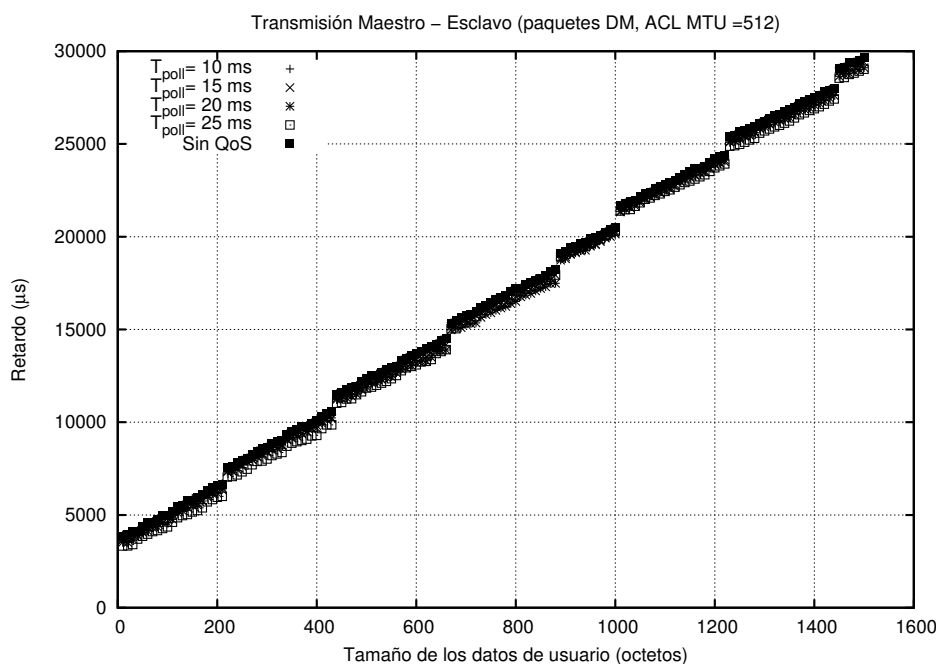


Figura 5.18.: Efecto de la configuración de la calidad de servicio, en el sentido de transmisión maestro-esclavo, para la versión de *firmware* 0xc5c para paquetes DM.

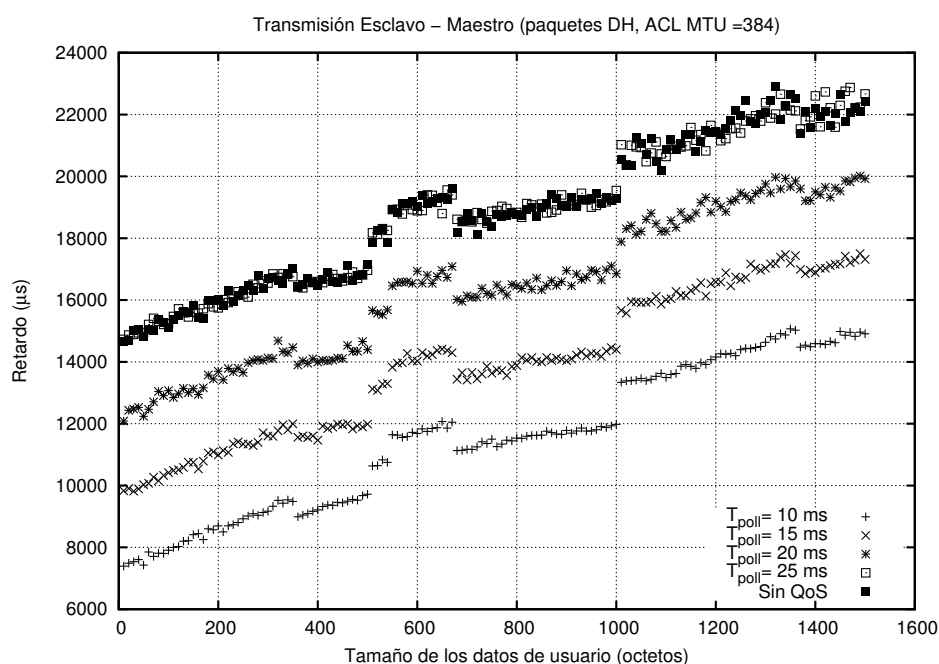
5.3.2.2. Sentido de transmisión esclavo-maestro

En el sentido esclavo-maestro, para la versión de *firmware* 0x7a6 y configurando el parámetro ACL MTU a un valor de 384 octetos, se han obtenido los resultados presentados en las gráficas de la figura 5.19, para todos los valores del parámetro *Latency* especificados en la tabla 5.3 y ambos tipos de paquetes.

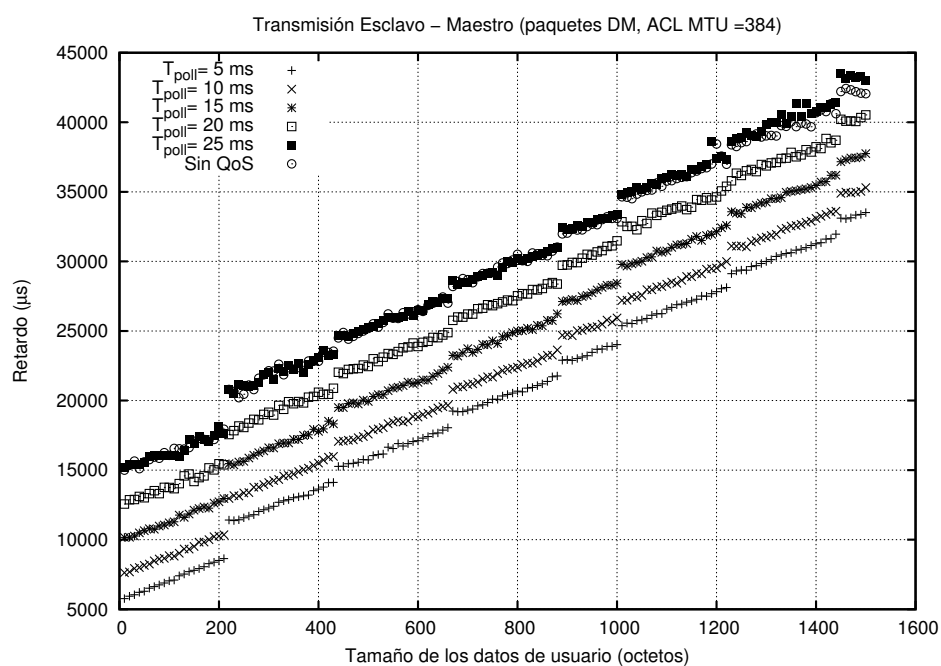
Por último, la repetición de las medidas llevada a cabo con la versión de *firmware* 0xc5c arroja los resultados presentados, a modo de ejemplo, en la figura 5.20, para el caso particular en el que el parámetro ACL MTU toma los valores de 1021 octetos (paquetes DH)

y de 512 octetos (paquetes DM), dado que, como se ha indicado en el apartado 5.3.1, no existe dependencia del valor de dicho parámetro.

De la observación de la citada figura se desprende que el efecto del valor del intervalo de *poll* configurado, T_{poll} (asimilable al parámetro *Latency*), es un incremento constante del retardo (reflejado en un desplazamiento vertical en las gráficas), tal y como también sucede en dispositivos del mismo fabricante de la versión Bluetooth 1.1. También, al igual que en dicha versión, y en el sentido maestro-esclavo puede constatarse que un intervalo de *poll* de 25 ms ($T_{poll} = 25$ ms), equivale a no configurar la calidad de servicio y que el grueso del *offset* asociado a cada intervalo de *poll* (T_{poll}) se corresponde con $T_{poll}/2$, aunque, como se explica en los siguientes apartados, a este valor se añade una componente introducida por la capa de transporte.

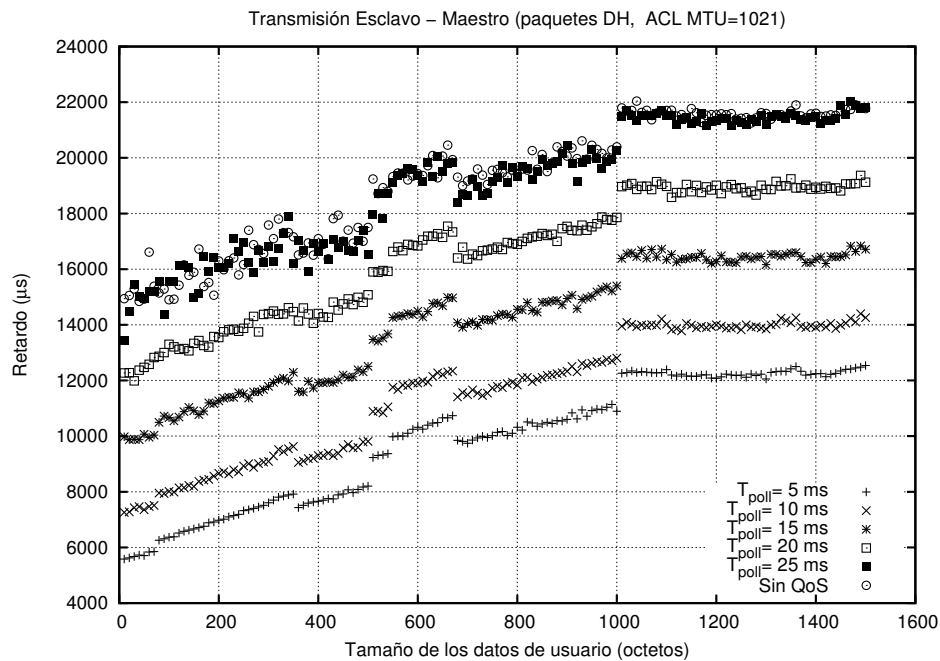


(a) Paquetes DH

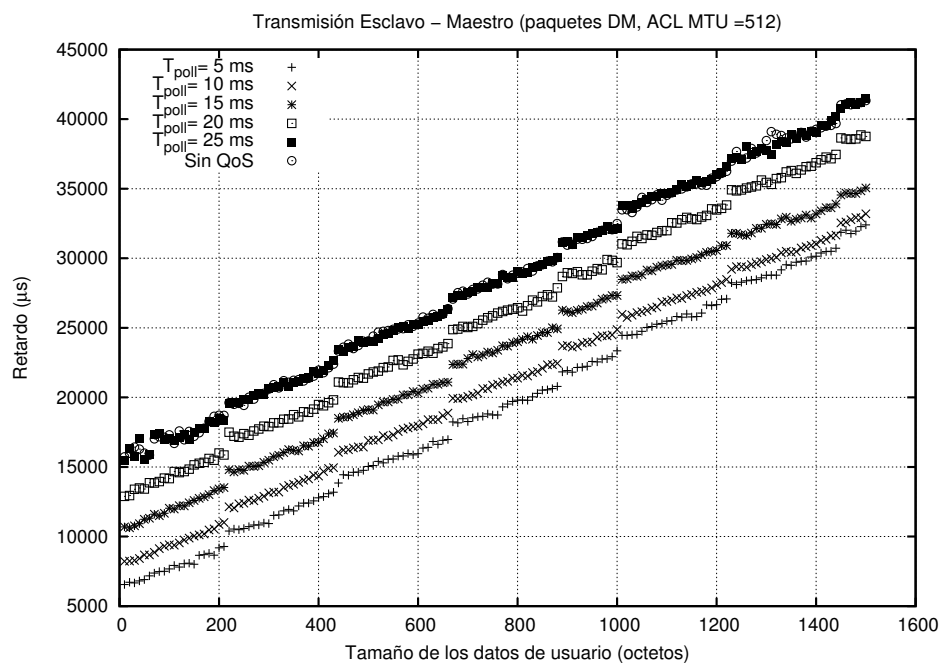


(b) Paquetes DM

Figura 5.19.: Efecto de la configuración de la calidad de servicio, en el sentido de transmisión esclavo-maestro, para la versión de *firmware* 0x7a6 y para un valor del parámetro ACL MTU de 384 octetos



(a) Paquetes DH y ACL MTU de 1021 octetos



(b) Paquetes DM y ACL MTU de 512 octetos

Figura 5.20.: Efecto de la configuración de la calidad de servicio, en el sentido de transmisión esclavo-maestro, para la versión de *firmware* 0xc5c

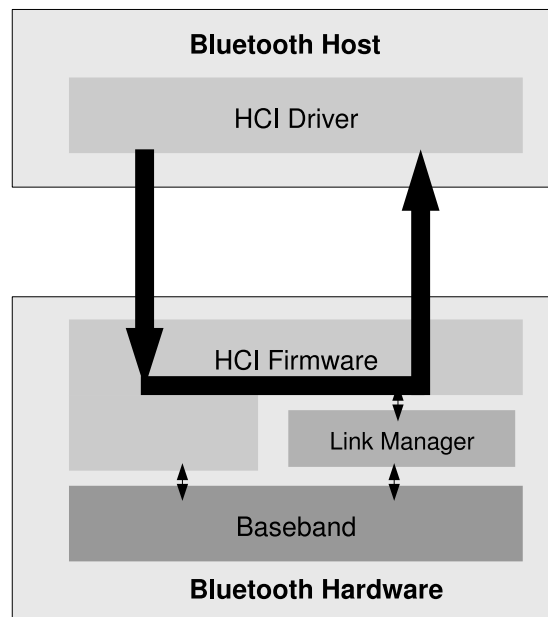
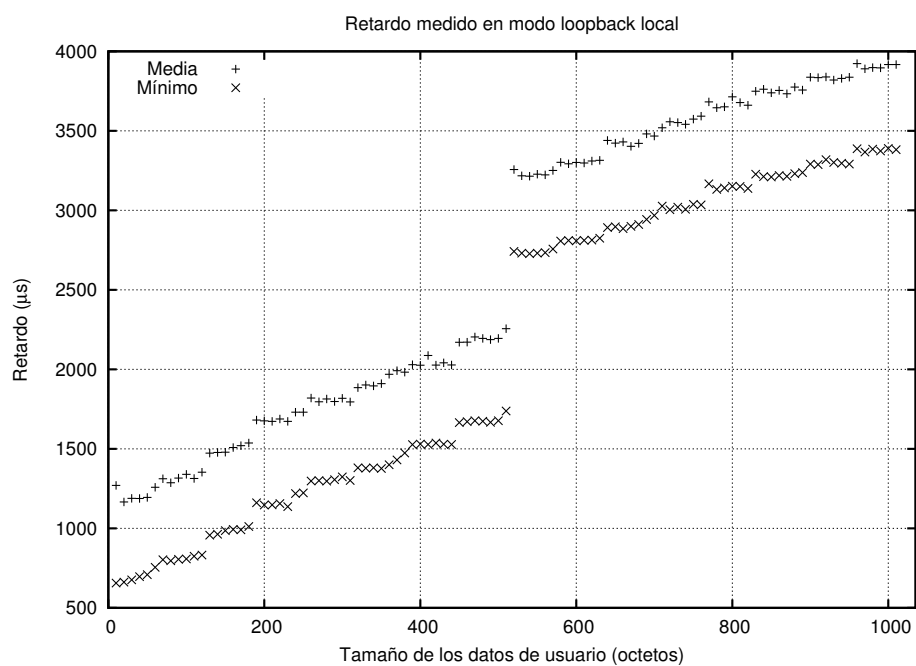
5.3.3. Caracterización de la capa de transporte

Buena parte de la divergencia entre el modelo teórico presentado en el capítulo anterior y las medidas realizadas se debe a que el modelo no incluye el retardo introducido por la capa de transporte HCTL. Para dicha capa de transporte, el estándar Bluetooth especifica las siguientes interfaces físicas: SD, UART-RS-232-C, *Three-wire* UART y USB [Bluetooth SIG04, Bluetooth SIG07, vol. 4], de las cuales, la interfaz USB es la soportada por los dispositivos utilizados y, en general, por la gran mayoría de los disponibles comercialmente. Concretamente, los dispositivos empleados, aunque conformes a la especificación USB 2.0 [USB.org00], sólo utilizan el modo *full-speed* (12 Mbps), compatible con USB 1.1, según el estándar [Bluetooth SIG04, Bluetooth SIG07, vol. 4].

Al respecto, cabe resaltar la importancia de analizar el comportamiento de la capa de transporte HCTL USB, dado que su aplicación no está limitada al estudio abordado en esta tesis con módulos Bluetooth externos. De hecho, incluso dispositivos con tecnología Bluetooth 2.0 + EDR integrada, como teléfonos móviles, incluyen una interfaz USB interna como capa HCTL para la comunicación con el módulo (*Host Controller*), debido a que, a diferencia de las restantes interfaces, proporciona un ancho de banda superior a la máxima tasa de transferencia del modo EDR.

La caracterización del retardo introducido por la capa de transporte HCTL USB se ha llevado a cabo utilizando el comando *HCI_Write_Loopback_Mode* para habilitar el modo *loopback* local, en el que el *Host Controller* devuelve al *Host* cada paquete de datos o de comandos generado por el *Host*, tal y como se ilustra en la figura 5.21a. Para ello, se ha desarrollado un programa en C, sobre la pila de protocolos BlueZ [BlueZ], que calcula el tiempo transcurrido desde el envío de paquetes ACL hasta que son devueltos por el *Host Controller*, para longitudes comprendidas entre 10 y 1010 octetos. Concretamente, para cada longitud se han realizado 100 repeticiones separadas un tiempo de 100 ms, midiéndose los retardos mínimo y medio, representados en la gráfica de la figura 5.21b. Nótese que existe un incremento del retardo de 1,20 ms en torno a los 512 octetos, delimitando claramente dos tramos con pendientes distintas.

La justificación de este comportamiento de la capa de transporte HCTL no es trivial, especialmente porque es un aspecto específico del funcionamiento del dispositivo Bluetooth

(a) Funcionamiento en el modo *loopback* local(b) Retardos mínimo y medio medido en el modo *loopback* local**Figura 5.21.:** Caracterización de la capa de transporte utilizando el modo *loopback* local

del fabricante CSR. Sin embargo, es plausible suponer que se trata del tamaño de un *buffer* intermedio asociado al transporte físico USB, lo que provoca una fragmentación interna al valor de 512 octetos, límite que, en lo sucesivo, se va a denotar como M_{USB} .

Con el fin de incorporar el efecto del transporte USB en el modelo teórico a partir del retardo medio, se propone la utilización de un modelo empírico que, empleando el método de mínimos cuadrados, proporcione una aproximación lineal por tramos adecuada al comportamiento registrado.

Así, se modela el retardo medio introducido por la capa de transporte, HCTL, en dispositivos del fabricante CSR, $t_{USB}(x)$, como la recta de regresión correspondiente a cada uno de los tramos indicados, resultando la siguiente ecuación empírica:

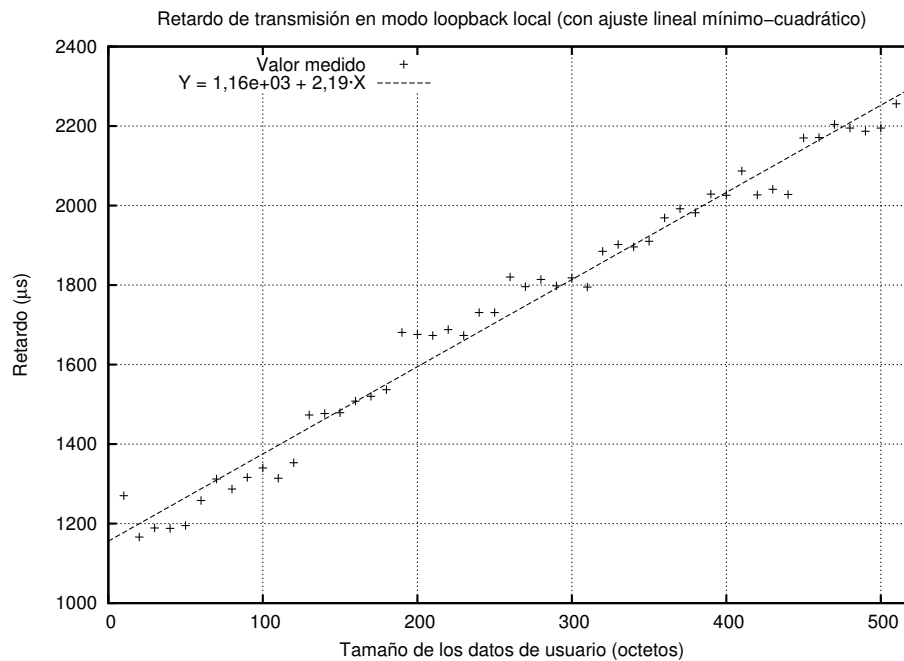
$$t_{USB}(x) = \begin{cases} 1,16 \cdot 10^3 + 2,19 \cdot x & x \leq M_{USB} \\ 2,39 \cdot 10^3 + 1,57 \cdot x & x > M_{USB} \end{cases} \quad (5.1)$$

La figura 5.22 representa el ajuste de mínimos cuadrados efectuado para ambos tramos. Por último, se ha constatado que el resultado es independiente del valor del parámetro ACL MTU configurado, tal y como puede observarse en la figura 5.23, en la que se muestra el retardo cuando dicho parámetro se configura a 1021 y 420 octetos.

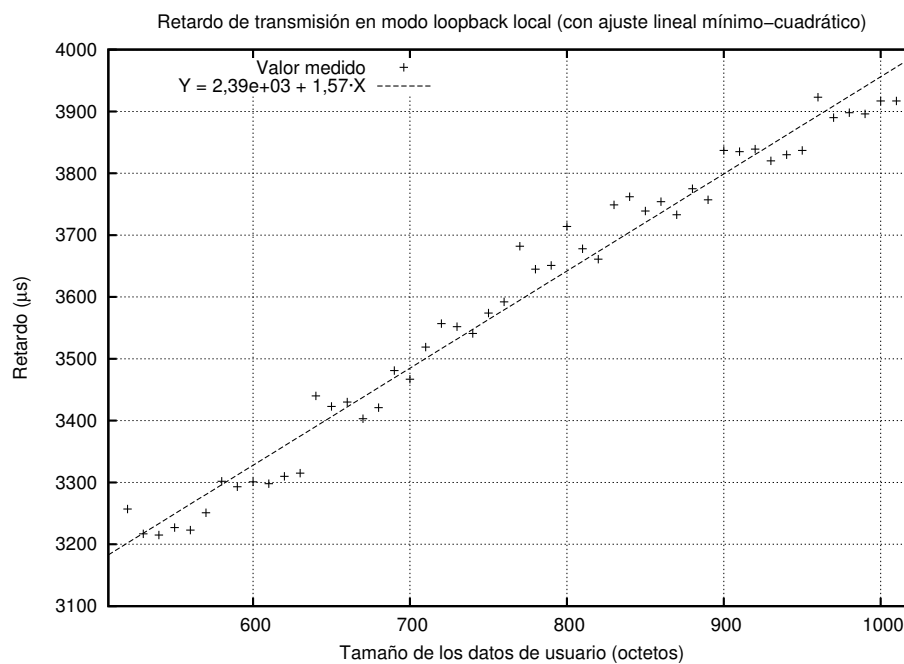
5.3.4. Modificación del modelo teórico para contemplar el efecto de la capa de transporte

En este apartado se corrige el modelo teórico presentado en el capítulo 4 para incorporar el efecto de la capa HCTL en el cálculo del retardo cuando se utilizan paquetes:

- DH en el modo EDR (véase el apartado 5.3.4.1).
- DM y DH en el modo básico (véase el apartado 5.3.4.2).



(a) Recta de regresión para longitudes inferiores a 512 octetos



(b) Recta de regresión para longitudes superiores a 512 octetos

Figura 5.22.: Ajuste lineal con mínimos cuadrados del retardo asociado a la capa de transporte HCTL

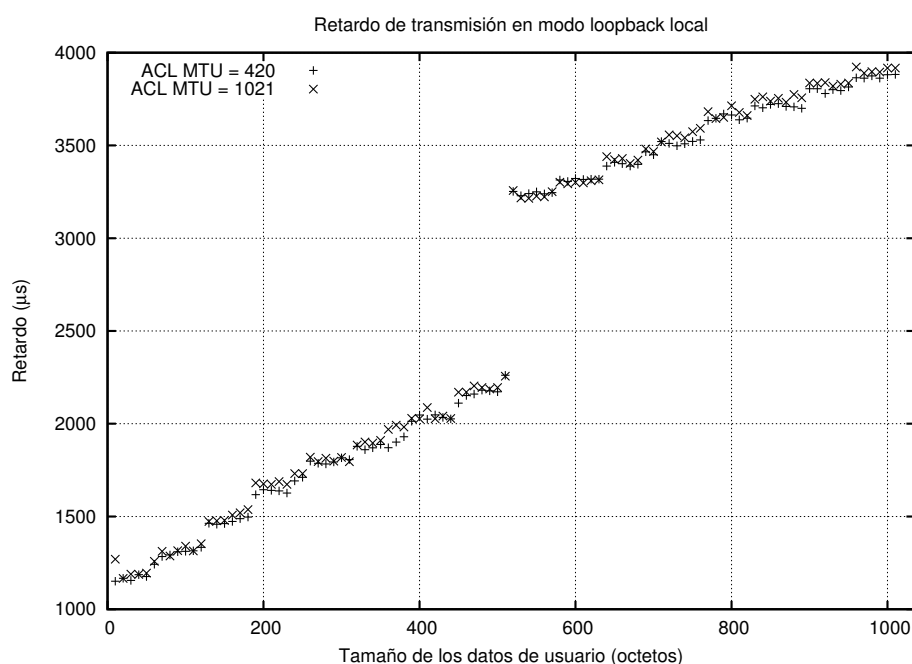


Figura 5.23.: Comparación del retardo medido en el modo *loopback* local para valores del valor del parámetro ACL MTU de 1021 y 420 octetos.

Las medidas utilizadas para contrastar el resultado de las correcciones que se proponen se han realizado con valores del parámetro ACL MTU que no influyen en el retardo. Así, se ha empleado una ACL MTU de 1021 octetos en el primer caso (paquetes DH en modo EDR), y de 512 octetos en el segundo caso (paquetes DH y DM del modo básico). Nótese que estos valores son superiores (o iguales) a las máximas capacidades de los paquetes: 3-DH5 (1021 octetos), en el modo EDR; y DM5 (224 octetos) y DH5 (339 octetos), en el modo básico.

En todos los casos, las medidas presentadas en esta sección se han ejecutado con la versión de *firmware* 0xc5c, que se caracteriza por tener un comportamiento más estable que la versión 0x7a6, además de ser la más extendida comercialmente¹. No obstante, a lo largo de la exposición se muestra cómo el ajuste del modelo es válido también para la versión 0x7a6. Adicionalmente, se recuerda que los resultados obtenidos son extrapolables a dispositivos con versiones posteriores de *firmware*, compatibles con Bluetooth 2.1 + EDR.

¹De los dispositivos adquiridos entre los años 2008 y 2009, un total de 24 unidades de distintos fabricantes (Ovislink, Conceptronic, Inves, Ansonic, Trust) integran la versión 0xc5c, frente a tan solo 4 unidades que se lograron encontrar con la versión 0x7a6.

En cuanto al orden de exposición, que comienza por la corrección del modelo para paquetes DH EDR, y no por DM o DH básicos, como se podría esperar, es conveniente aclarar que refleja fielmente el procedimiento de investigación seguido, ya que éste, desde el principio, se ha centrado en el núcleo de la gran aportación de Bluetooth 2.0 + EDR: el modo EDR.

5.3.4.1. Paquetes DH en el modo EDR

La inclusión en el modelo teórico del efecto de la capa de transporte supone únicamente sumar el retardo introducido por esta capa al previsto teóricamente, incluyéndolo en cada una de las componentes básicas del retardo, según se describe a continuación:

$$t_{TX}^c(x) = t_{TX}(x) + C_{TX}^{USB}(x) \quad (5.2)$$

$$t_{ACK}^c(x) = t_{ACK}(x) + C_{TX}^{USB}(x) \quad (5.3)$$

donde el superíndice c de las componentes $t_{TX}^c(x)$ y $t_{ACK}^c(x)$ indica que las originales, $t_{TX}(x)$ y $t_{ACK}(x)$, han sido modificadas para incluir el efecto de la capa de transporte, representado por el factor de corrección $C_{TX}^{USB}(x)$ que, en este caso, se corresponde directamente con el retardo consumido en la transferencia USB de los x octetos a transmitir vía Bluetooth:

$$C_{TX}^{USB}(x) = t_{USB}(x) \quad (5.4)$$

A partir de estas expresiones se calcula el retardo para el perfil SPP en función de las componentes $t_{TX}^c(x)$ y $t_{ACK}^c(x)$, de acuerdo con la siguiente ecuación:

$$\begin{aligned} \overline{t_R}(x) = & t_{off} + n_{ff}(x) \cdot t_{ACK}^c(M(L_R)) + \\ & + t_{TX}^c(M(L_{ff}(x))) \end{aligned} \quad (5.5)$$

donde el término designado como t_{off} representa un *offset* promedio cuyo valor se va a determinar a posteriori, después de haber modelado el retardo introducido por la capa de transporte, y teniendo en cuenta que, en el sentido de transmisión esclavo-maestro, ha de incluir el *slot* en el que maestro direcciona al esclavo.

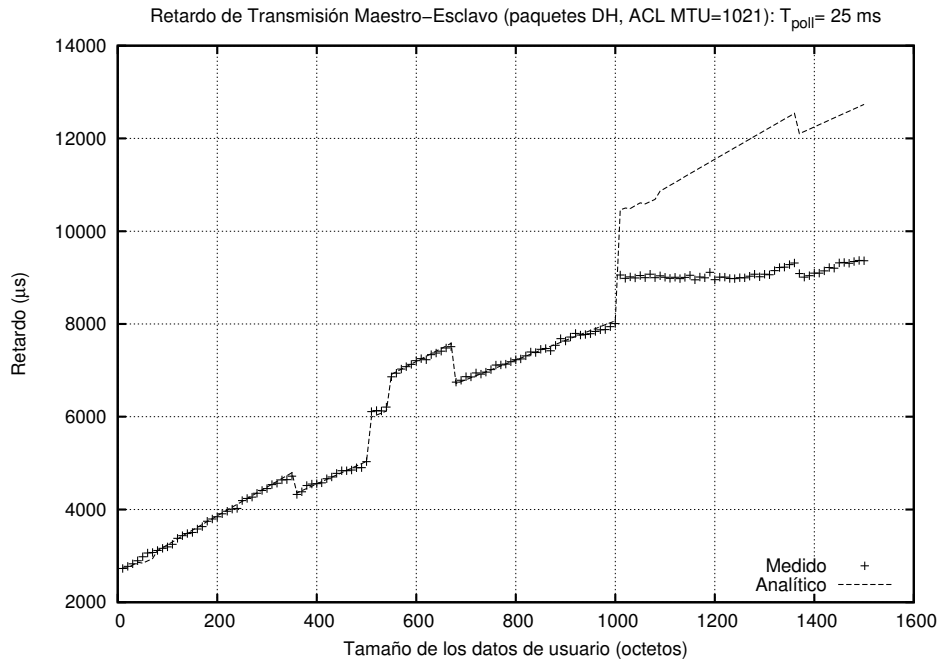


Figura 5.24.: Comparación del retardo medido y el teórico incluyendo el retardo introducido por la capa de transporte USB, en el sentido maestro-esclavo, para un valor del parámetro ACL MTU de 1021 octetos (Revisión de HCI: 0xc5c)

Nótese que, en la ecuación 5.5, el término $t_{ACK}^c(x)$ toma siempre un valor constante, resultante de sumar al tiempo de transmisión y confirmación, $t_{ACK}(M(L_R)) = 6 \cdot T_S$, el retardo que introduce la capa de transporte: $t_{USB}(M(L_R))$. Tal y como puede observarse en la gráfica de la figura 5.24, el ajuste del modelo modificado es muy acertado hasta que se excede el valor de la MTU de L2CAP para RFCOMM, $M_R = 1013$ octetos ($L_R = 1008$ octetos). A partir de este punto, el retardo modelado es superior al medido. Al respecto, cabe destacar que los 1013 octetos de la trama RFCOMM, sumados a los 4 octetos correspondientes a la cabecera añadida por el nivel L2CAP, pueden ser transportados en un único paquete de 5 slots de velocidad máxima (3-DH5), al no excederse su capacidad (1021 octetos).

Adicionalmente, las figuras 5.25a y 5.25b, para el sentido de transmisión esclavo-maestro, muestran la comparación del retardo previsto teóricamente y las medidas obtenidas para la versión de *firmware* 0xc5c, para un intervalo de *poll* (T_{poll}) de 10 ms y 20 ms. La razón de haber utilizado valores distintos a 25 ms para el intervalo de *poll* en el sentido esclavo-maestro es discriminar el *offset* que se atribuye al efecto de la configuración de

la calidad de servicio del retardo introducido por la capa de transporte. Por consiguiente, dado que el incremento del retardo en el sentido esclavo-maestro sí depende del valor del parámetro T_{poll} configurado, a diferencia de lo que sucede en el sentido maestro-esclavo, ha sido necesario definir el término t_{off} en función del sentido de transmisión:

$$t_{off} = \begin{cases} 2 \cdot T_S & \text{Sentido maestro-esclavo} \\ \frac{T_{poll}}{2} + T_S & \text{Sentido esclavo-maestro} \end{cases} \quad (5.6)$$

La expresión anterior refleja la asimetría del funcionamiento del mecanismo de *poll* en el caso de dispositivos Bluetooth 2.0 + EDR. El esclavo se ve obligado a esperar que el maestro de la *piconet* lo dirija mediante el mecanismo de *poll*, de promedio cada $T_{poll}/2$, incrementando, por tanto, el retardo de transmisión de datos en esta cantidad. Por su parte, el maestro, que gestiona el mecanismo de *poll*, no espera un tiempo variable para direccionar al esclavo ya que los resultados indican que el retardo adicional es fijo.

A modo de inciso, en la figura 5.26 se puede constatar que el modelo también responde al comportamiento de la versión de *firmware* 0x7a6. Según se ha anticipado en el apartado 5.3.1, el retardo obtenido para esta versión coincide en todos los casos, es decir, para todos los posibles valores del parámetro ACL MTU, con la medida obtenida para la versión de *firmware* 0xc5c en el caso particular de la ACL MTU de 1021 octetos, diferenciándose en que presenta una mayor dispersión y en el *offset* que, para el intervalo de *poll* utilizado, $T_{poll} = 25 \text{ ms}$ (40 *slots*), es de $t_{off} = 30/2 \text{ slots}$, siendo 30 *slots*, un valor inferior al intervalo de *poll* configurado. Este resultado indica que, a diferencia de la versión 0xc5c, el maestro sí está condicionado por el mecanismo de *poll*, aunque en menor medida que el esclavo, ya que el *offset* no se corresponde, en ningún caso, con $T_{poll}/2$.

Una vez hecha esta aclaración relativa a la versión de *firmware* 0x7a6, puede concluirse que la inclusión del modelo de la capa de transporte en el cálculo teórico responde al retardo obtenido empíricamente hasta que se excede el valor de la MTU de L2CAP para RFCOMM, $M_R = 1013$ octetos ($L_R = 1008$ octetos), límite inferior a la máxima capacidad de un paquete de 5 *slots*, 3-DH5. Tal y como se apunta en el siguiente apartado, la causa de la sobreestimación está relacionada con el modo en que la ecuación 5.5 computa el retardo introducido por la capa de transporte en la transmisión del fragmento RFCOMM

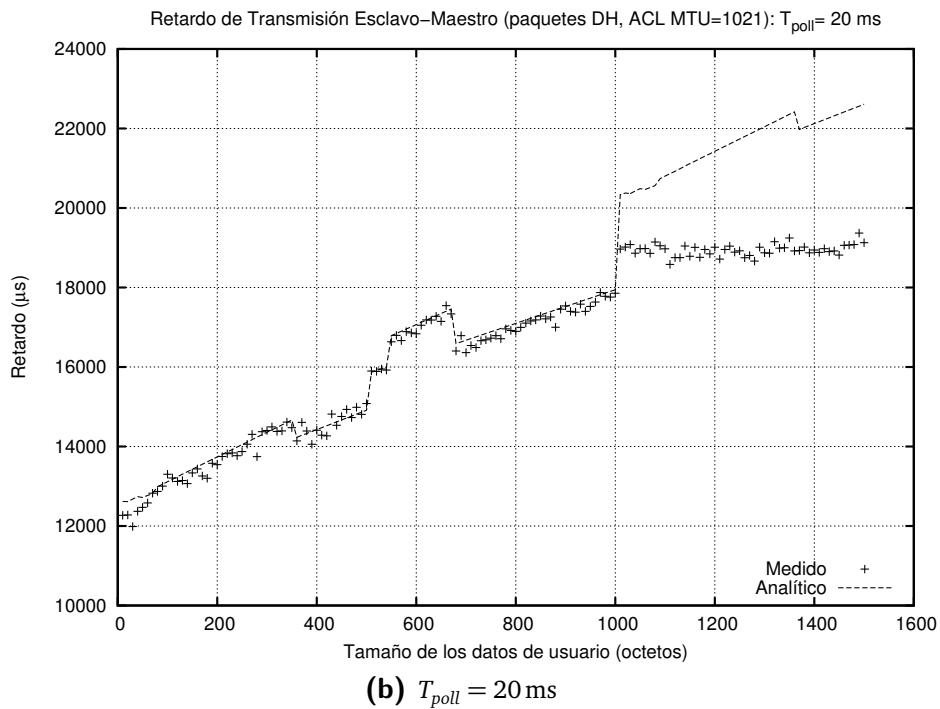
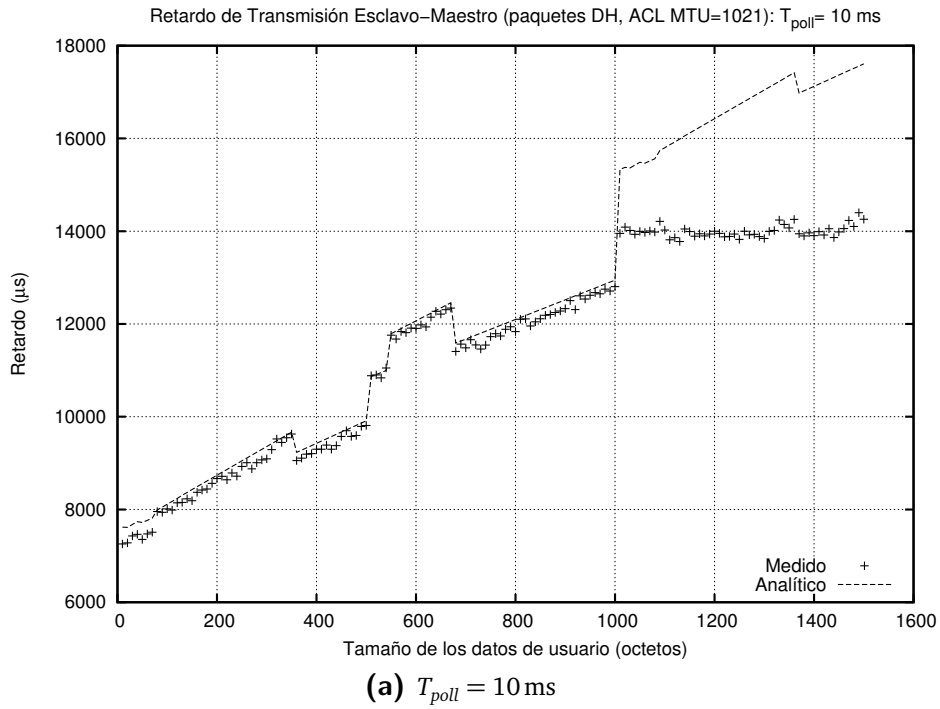


Figura 5.25.: Comparación del retardo medido y el teórico incluyendo el retardo introducido por la capa de transporte USB, en el sentido esclavo-maestro, para paquetes DH y un valor del parámetro ACL MTU de 1021 octetos, con corrección para longitudes superiores a $M_R = 1013$ octetos (Revisión de HCI: 0xc5c)

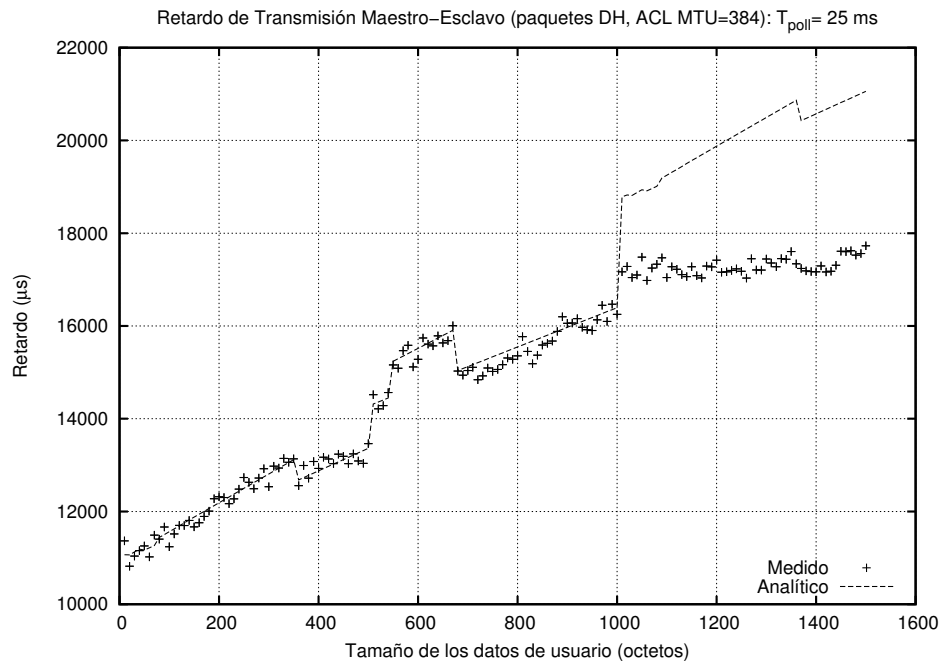


Figura 5.26.: Comparación del retardo medido y el teórico incluyendo el retardo introducido por la capa de transporte USB, en el sentido maestro-esclavo, con corrección para longitudes superiores a $M_R = 1013$ octetos, para un valor del parámetro ACL MTU de 384 octetos (Revisión de HCI: 0x7a6)

inicial, mediante la componente $t_{ACK}^c(x)$. La razón es que no se está teniendo en cuenta que, probablemente, mientras que este primer fragmento es transmitido vía radio, el segundo fragmento esté siendo transferido por el *Host* al dispositivo y, análogamente, la entrega al *Host* (por parte del dispositivo receptor) del primer fragmento recibido tenga lugar al mismo tiempo que la transmisión Bluetooth del segundo fragmento. No obstante, la identificación del origen de la sobreestimación y, también, posibles correcciones, se analizan en los apartados 5.3.5 y 5.3.6, al evaluar y modelar la combinación de los efectos debidos a la capa de transporte USB y a la fragmentación realizada al valor del parámetro ACL MTU en el sentido de transmisión maestro-esclavo.

5.3.4.2. Paquetes DM y DH en el modo básico

A continuación se expone cómo ha de adaptarse el modelo para el caso en el que se configura el dispositivo para excluir el uso de paquetes DH del modo EDR. Para ello

se parte de la formulación definida en el apartado anterior de forma que el retardo de transmisión a nivel de RFCOMM, en el sentido de transmisión maestro-esclavo, se determina mediante la ecuación 5.5, a partir de las componentes $t_{TX}^c(x)$ y $t_{ACK}^c(x)$ definidas por las ecuaciones 5.2 y 5.3, en las que el término $C_{TX}^{USB}(x)$, que representa la corrección introducida para modelar la influencia sobre el retardo de transmisión de la transferencia a través de la capa de transporte HCTL, ha de ser redefinido de acuerdo con la siguiente ecuación:

$$C_{TX}^{USB}(x) = \begin{cases} \frac{1}{2}t_{USB}(x) & 0 \leq x \leq M_{USB} \\ \frac{1}{2}t_{USB}(0) + \frac{1}{2}t_{USB}(x - M_{USB}) & (x > M_{USB}) \end{cases} \quad (5.7)$$

Nótese que, en este caso, dependiendo de que x sea inferior o no al límite impuesto por la capa de transporte USB, se suma al retardo de transmisión Bluetooth de x octetos, $t_{TX}(x)$ y $t_{ACK}(x)$, un factor variable:

- $x \leq M_{USB}$: La mitad del retardo introducido por la capa de transporte para ese mismo número de octetos.
- $x > M_{USB}$: Al ser la longitud de los datos a transferir por la interfaz USB superior al límite M_{USB} , se produce una fragmentación que da lugar a un bloque de 512 octetos y a un segundo bloque de $x - M_{USB}$ octetos. Por tanto, se suman dos contribuciones:
 1. Componente constante, $\frac{1}{2}t_{USB}(0)$, asociada al primer fragmento de 512 octetos (M_{USB}).
 2. Componente dependiente de la longitud, $\frac{1}{2}t_{USB}(x - M_{USB})$, correspondiente al residuo de $x - M_{USB}$ octetos, que no excede los 512 octetos (M_{USB}).

En las figuras 5.27a y 5.27b se compara el retardo calculado de acuerdo con la ecuación 5.5 con las medidas realizadas con dispositivos Bluetooth 2.0 + EDR con la versión de *firmware* 0xc5c, en el sentido de transmisión maestro-esclavo, sin haber configurado la calidad de servicio, y utilizando, respectivamente, paquetes DM y paquetes DH (excluyendo los paquetes DH EDR).

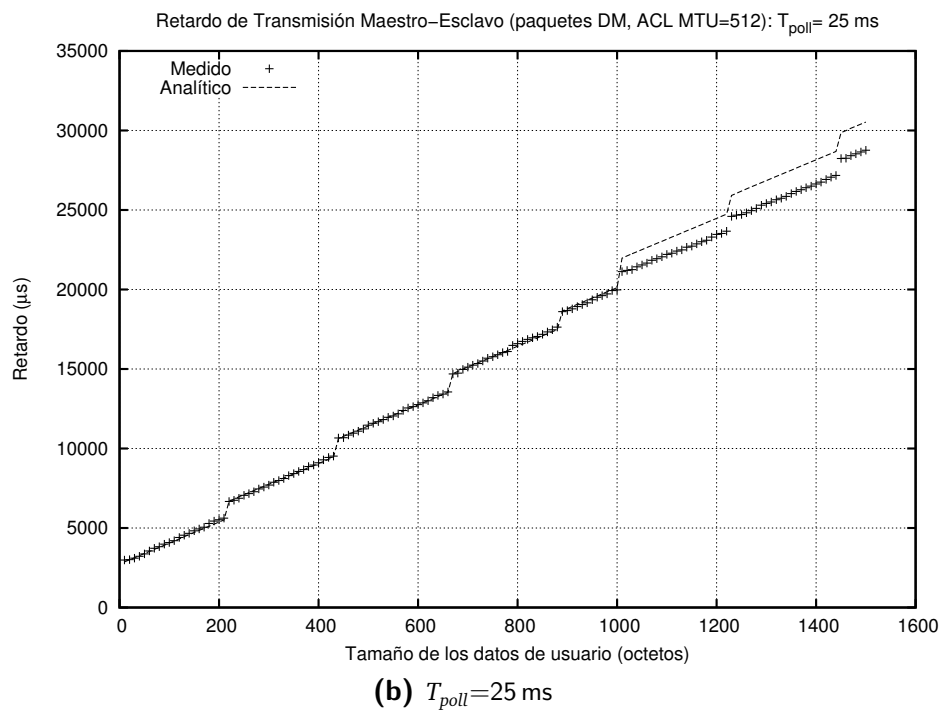
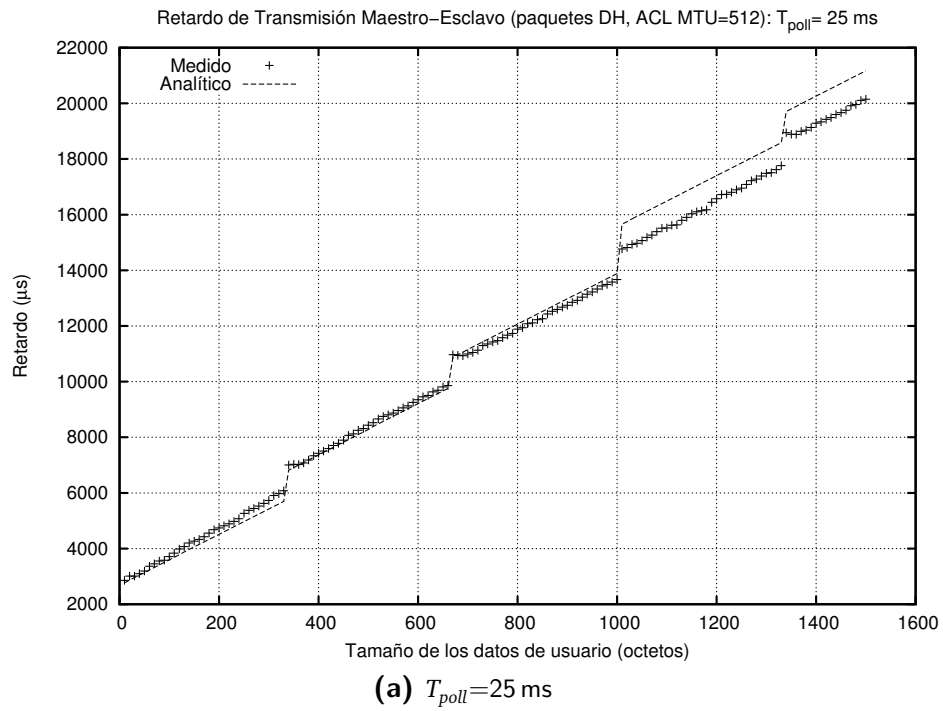


Figura 5.27.: Comparación del retardo medido y el teórico, incluyendo el retardo introducido por la capa de transporte USB, en el sentido maestro-esclavo, para paquetes DH y DM (Revisión de HCI: 0xc5c)

El valor de la constante t_{off} con la que se ha obtenido la curva teórica, correspondiente a la ecuación 5.5, representada en ambas figuras para el sentido maestro-esclavo, no depende del valor del parámetro T_{poll} configurado, y se ha podido constatar que el intervalo entre direccionamientos consecutivos al esclavo es de 6 slots, tanto para los paquetes DH como DM, lo que se traduce en un *offset* promedio de $t_{off} = 3 \cdot T_S$. Adicionalmente, en lo que respecta al sentido esclavo-maestro, el valor de esta constante no resulta ser independiente del intervalo de *poll* configurado, de forma que, aunque el grueso del *offset* corresponde a $T_{poll}/2$, tras computar el retardo introducido por la capa de transporte y el slot de direccionamiento, es necesario sumar dos slots adicionales ($2 \cdot T_S$). Así pues, la expresión que define el término t_{off} es la siguiente:

$$t_{off} = \begin{cases} 3 \cdot T_S & \text{Sentido maestro-esclavo} \\ \frac{T_{poll}}{2} + 3 \cdot T_S & \text{Sentido esclavo-maestro} \end{cases} \quad (5.8)$$

En las gráficas de las figuras 5.27a y 5.27b puede apreciarse que, al igual que para el modo EDR, se produce una sobreestimación cuando los datos de usuario exceden el tamaño la MTU de L2CAP para RFCOMM, $M_R = 1013$ octetos ($L_R = 1008$ octetos). Sin embargo, a diferencia de lo que sucede en el modo EDR, este límite no coincide con la máxima capacidad de un paquete de 5 slots. Estos resultados, de hecho, proporcionan un indicio acerca del motivo por el que tiene lugar la sobreestimación: una vez que se exceden las capacidades de un paquete DH5 y DM5, la componente $t_{TX}(x)$ hace uso, en su rama recursiva, de la componente $t_{ACK}(x)$ básica (ecuación 4.11), que no incluye ninguna corrección para introducir el retardo asociado a la capa de transporte. Por contra, al superarse los 1013 octetos, puede observarse en la ecuación 5.5 que, para computar el retardo del primer fragmento RFCOMM, interviene la componente $t_{ACK}^c(x)$ que, al retardo de transmisión (incluyendo el slot de confirmación), añade $t_{USB}(x)$.

Por último, es conveniente aclarar que las gráficas que comparan los retardos para la versión 0x7a6 y, en el sentido esclavo-maestro, para la versión 0xc5c, se han incluido en el apartado 5.3.6.1, una vez corregida la sobrestimación que tiene lugar a partir de los 1008 octetos de datos de usuario.

5.3.5. Modificación del modelo teórico para contemplar el efecto del transporte USB y la fragmentación realizada al valor del parámetro ACL MTU en el sentido de transmisión maestro-esclavo

En esta sección se describe cómo ha de adaptarse el modelo para reflejar la dependencia respecto del valor del parámetro ACL MTU, detectada en el sentido maestro-esclavo a partir de la versión de *firmware* 0xc5c, que se pone de manifiesto tanto en el modo básico como en el modo EDR cuando se configura a un valor inferior a la máxima capacidad de un paquete de 5 slots. La corrección del modelo se va a abordar en los siguientes pasos:

1. En el apartado 5.3.5.1 se incorpora en las componentes básicas del retardo de transmisión teniendo el efecto del parámetro ACL MTU.
2. En el apartado 5.3.5.2 se lleva a cabo la redefinición de los componentes básicas del retardo de transmisión para englobar la suma de los efectos de la fragmentación conforme al valor del parámetro ACL MTU y del retardo introducido por la capa de transporte.
3. En el apartado 5.3.5.3 se reformula la ecuación del cálculo del retardo de transmisión a nivel RFCOMM.

Por último, una vez completadas las correcciones, en el apartado 5.3.5.4 se presentan los resultados obtenidos para el modo básico y para el modo EDR.

5.3.5.1. Cálculo de las componentes básicas del retardo de transmisión teniendo en cuenta el efecto del parámetro ACL MTU

Para incluir el efecto de la fragmentación según el valor del parámetro ACL MTU sobre la transmisión a nivel BB se van a definir dos componentes $t_{ACK}^{ACL}(x)$ y $t_{TX}^{ACL}(x)$, basadas en las

originales, t_{ACK} y $t_{TX}(x)$, según se indica a continuación:

$$t_{ACK}^{ACL}(x) = \begin{cases} t_{ACK}(x) & 0 \leq x \leq M_{BB} \\ t_{ACK}(M_{BB}) + t_{ACK}(x - M_{BB}) & \text{en otro caso} \end{cases} \quad (5.9)$$

$$t_{TX}^{ACL}(x) = \begin{cases} t_{TX}(x) & 0 \leq x \leq M_{BB} \\ t_{ACK}(M_{BB}) + t_{TX}(x - M_{BB}) & \text{en otro caso} \end{cases} \quad (5.10)$$

donde M_{BB} representa el mínimo entre el parámetro ACL MTU, M_{ACL} , y la máxima capacidad de un paquete del nivel BB de 5 slots, L_5 , que será 224 octetos y 339 octetos, para los paquetes DM5 y DH5 del modo básico, y 1021 octetos, para el paquete 3-DH5 del modo EDR:

$$M_{BB} = \text{mín}(M_{ACL}, L_5) \quad (5.11)$$

Nótese que las ecuaciones 5.9 y 5.10 calculan el retardo de transmisión de los datos del nivel L2CAP, computando el slot de confirmación ($t_{ACK}^{ACL}(x)$) y sin computarlo ($t_{TX}^{ACL}(x)$), de forma diferente dependiendo de la relación entre su longitud y el mínimo entre dos valores, el parámetro ACL MTU (M_{ACL}) y la máxima longitud de un paquete BB de 5 slots (L_5), M_{BB} , según se indica a continuación:

- Longitud inferior a M_{BB} octetos: El retardo previsto originalmente, t_{ACK} y $t_{TX}(x)$, obtenido, respectivamente, mediante las ecuaciones 4.11 y 4.13.
- Longitud superior a M_{BB} octetos: Se estima que es la suma de dos componentes:
 1. El retardo de transmisión de M_{BB} octetos, considerando la confirmación del esclavo: $t_{ACK}(M_{BB})$, tanto si se computa $t_{ACK}^{ACL}(x)$ como $t_{TX}^{ACL}(x)$.
 2. El retardo de transmisión de los octetos restantes $-(x - M_{BB})-$, calculado utilizando las componentes básicas, $t_{ACK}(x - M_{BB})$ y $t_{TX}(x - M_{BB})$, según se trate de $t_{ACK}^{ACL}(x)$ ó de $t_{TX}^{ACL}(x)$.

Cabe destacar que la fragmentación de las tramas L2CAP conforme al valor del parámetro ACL MTU sólo afecta a la transmisión Bluetooth del primero de los fragmentos entregados por el *Host* al dispositivo cuando su valor se configura por debajo de la máxima capacidad de un paquete de 5 *slots*.

A modo de ejemplo, para la transferencia de 678 octetos de datos de usuario, en la figura 5.28a se muestra cómo al tomar el parámetro M_{ACL} el valor de 310 octetos se emplean 3 paquetes BB (DH5, DH5 y DH3), mientras que si M_{ACL} se fija a 350 octetos, los datos pueden enviarse en dos paquetes BB, tal y como se observa en la figura 5.28b.

Por último, cabe señalar que el efecto de la fragmentación de acuerdo con el valor del parámetro ACL MTU sobre la transferencia a través de la capa de transporte se analiza en el siguiente apartado.

5.3.5.2. Cálculo de las componentes básicas del retardo de transmisión teniendo en cuenta el efecto del parámetro ACL MTU y de la transferencia USB

Los resultados presentados en el apartado 5.3.4.2, ponen de manifiesto que la sobreestimación que tiene lugar al superarse los 1013 octetos ($L_R = 1008$ octetos), es debida a que la ecuación 5.5, para computar el retardo de un fragmento RFCOMM inicial (no único), utiliza la componente $t_{ACK}^c(x)$ que, al retardo de transmisión (incluyendo el *slot* de confirmación), añade el tiempo introducido por la capa de transporte USB, $C_{TX}^{USB}(x)$. La justificación se encuentra al observar el retardo correspondiente a las longitudes comprendidas entre el límite en el que se exceden las capacidades de un paquete DH5 y DM5 (339 octetos ó 224 octetos) y los 1013 octetos. En este rango de longitudes, para el cual el retardo previsto teóricamente se ajusta al medido empíricamente, cabe destacar que, según se ha explicado en el apartado 5.3.4.2, la componente $t_{TX}(x)$ hace uso, en su rama recursiva, de la componente $t_{ACK}(x)$ básica (ecuación 4.11), que no incluye ninguna corrección para introducir el retardo asociado a la capa de transporte. Así pues, partiendo de este supuesto se va a modificar la definición del término $t_{ACK}^c(x)$ utilizada por la ecuación 5.5 para obtener el retardo a nivel RFCOMM cuando la longitud de los datos de usuario exceden los 1008 octetos.

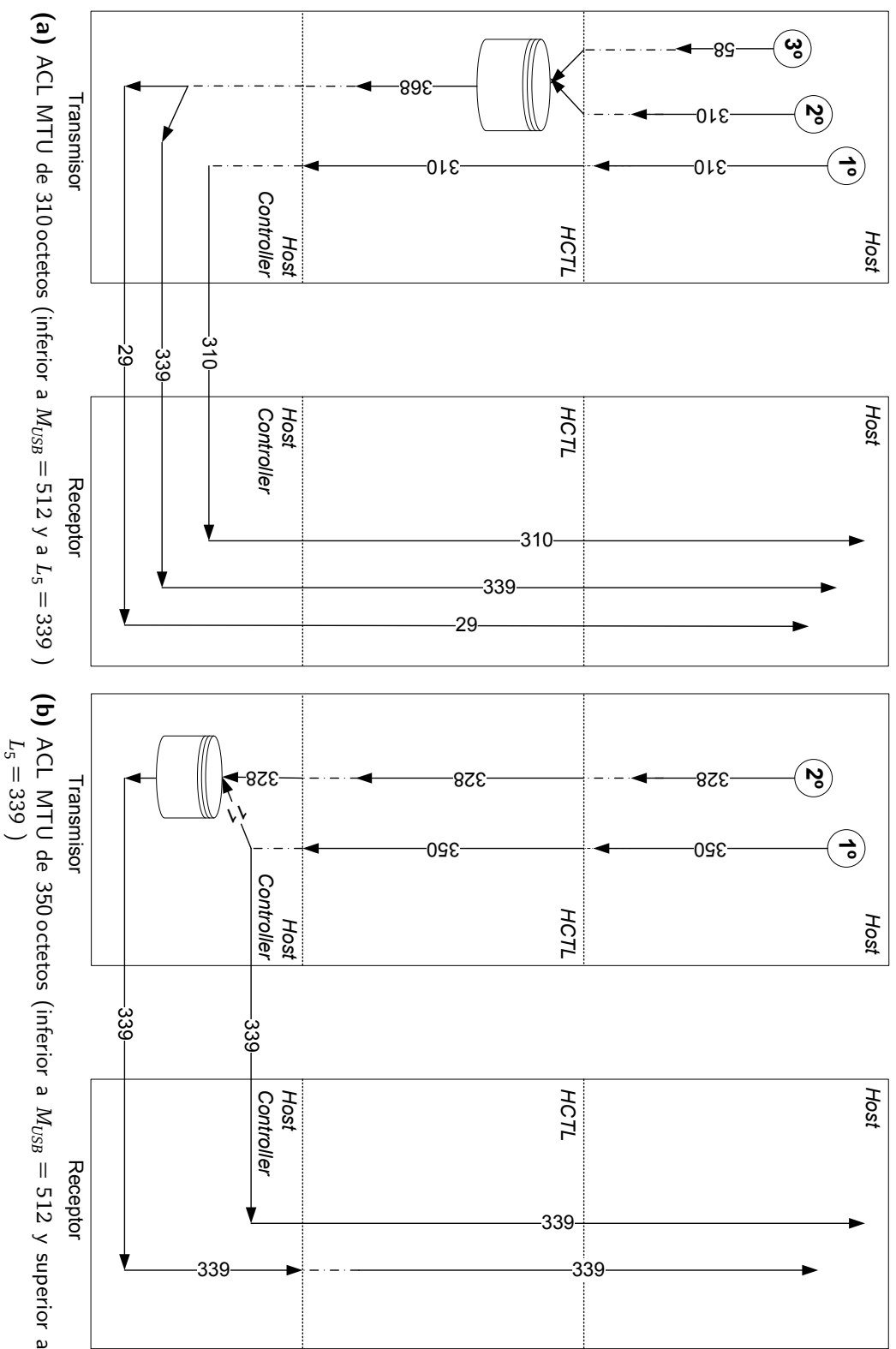


Figura 5.28.: Efecto de la fragmentación al tamaño del parámetro ACL MTU y de la capa de transporte para la transferencia de una trama L2CAP que encapsula una trama de RFCOMM con 678 octetos de datos de usuario, respectivamente.

Por consiguiente, el término $t_{ACK}^c(x)$ va a ser redefinido a fin de no incluir el efecto de la transferencia USB, coincidiendo por tanto su definición con la componente $t_{ACK}^{ACL}(x)$ que sí tiene en cuenta la influencia del parámetro ACL MTU:

$$t_{ACK}^c(x) = t_{ACK}^{ACL}(x) \quad (5.12)$$

Adicionalmente, el término $t_{TX}^c(x)$, definido originalmente por la ecuación 5.2 en la sección 5.3.4.1, ha de ser reformulado para englobar la suma de los efectos de la fragmentación conforme al valor del parámetro ACL MTU y del retardo introducido por la capa de transporte, de acuerdo con la siguiente expresión:

$$t_{TX}^c(x) = t_{TX}^{ACL}(x) + C_{TX}^{USB}(x) \quad (5.13)$$

En la ecuación anterior el término $C_{TX}^{USB}(x)$ representa, al igual que en la sección anterior, la corrección introducida para modelar la influencia de la transferencia a través de la capa de transporte sobre el retardo de transmisión con el matiz adicional de que en este caso, además, hay que tener en cuenta que su efecto depende de la fragmentación que se produce de acuerdo con el valor del parámetro ACL MTU. Su definición, que difiere para cada uno de los modos de funcionamiento, se lleva a cabo en los siguientes epígrafes.

Paquetes DH en el modo EDR

La componente $C_{TX}^{USB}(x)$, representa el retardo introducido por la capa de transporte, considerando también la dependencia respecto del parámetro ACL MTU, y viene dada por la siguiente ecuación:

$$C_{TX}^{USB}(x) = \begin{cases} t_{USB}(x) & 0 \leq x \leq M_{ACL} \\ \frac{1}{2}t_{USB}(\min(M_{ACL}, M_{USB})) + & x - M_{ACL} \leq M_{USB} \\ + \frac{1}{2}t_{USB}(x - \min(M_{ACL}, M_{USB})) & \\ \frac{1}{2}t_{USB}(M_{USB}) + t_{USB}(M_{ACL}) & \text{en otro caso} \end{cases} \quad (5.14)$$

Nótese que el término $C_{TX}^{USB}(x)$ representa la corrección introducida para modelar el efecto de la fragmentación de la trama *B-Frame* de L2CAP al tamaño dado por el parámetro ACL MTU para su transferencia por la interfaz HCI, considerando el retardo asociado a la capa de transporte. Concretamente, en la ecuación 5.14, se computa el retardo introducido por USB, diferenciándose tres casos, atendiendo a la relación entre el tamaño de la trama L2CAP (correspondiente al argumento x), y los parámetros ACL MTU (M_{ACL}) y el límite impuesto por la capa de transporte USB (M_{USB}):

- El tamaño de la trama L2CAP es inferior o igual al valor del parámetro ACL MTU ($x \leq M_{ACL}$): Se transmite un único fragmento, de forma que tan solo se añade el retardo introducido por la capa de transporte para esa longitud: $t_{USB}(x)$
- La diferencia entre el tamaño de la trama L2CAP y el valor del parámetro ACL MTU no excede los 512 octetos ($x - M_{ACL} \leq M_{USB}$): En este caso, se transfieren dos fragmentos, sumándose dos componentes:
 1. Componente constante, $\frac{1}{2}t_{USB}(\min(M_{ACL}, M_{USB}))$, asociada al primer fragmento, cuya longitud es el mínimo de dos valores: ACL MTU (M_{ACL}) y 512 octetos (M_{USB}), ya que, según se ha podido comprobar al caracterizar la capa de transporte, la transferencia por la capa de transporte está limitada a bloques de 512 octetos.
 2. Componente dependiente de la longitud, $\frac{1}{2}t_{USB}(x - \min(M_{ACL}, M_{USB}))$, correspondiente al residuo que, dada la condición ($x - M_{ACL} \leq M_{USB}$), no va a exceder los 512 octetos (M_{USB}).

Obsérvese que sólo contribuye la mitad del tiempo consumido en las transferencias USB. La razón es que al tratarse de dos fragmentos que se envían secuencialmente, sucede que la transmisión del segundo y la recepción del primero tienen lugar de forma simultánea.

- Restantes casos: $\frac{1}{2}t_{USB}(M_{USB}) + t_{USB}(M_{ACL})$. En este caso son tres, al menos, los fragmentos que se envían secuencialmente. En concreto, el primer fragmento de tamaño ACL MTU se transmite independientemente. Sin embargo, el segundo, también, de tamaño ACL MTU, y los sucesivos (de $x - n \cdot M_{ACL}$ octetos), se reensamblan para su transmisión en paquetes BB. Por esta razón, a diferencia del caso anterior, se añade un

retardo adicional de valor $\frac{1}{2}t_{USB}(M_{ACL})$, ya que no se paraleliza la transferencia por USB del tercer fragmento con la recepción del segundo, al recombinarse para su envío vía radio.

Paquetes DM y DH en el modo básico

Para el modo básico, la componente $C_{TX}^{USB}(x)$, segundo término de la ecuación 5.13, al igual que en el caso anterior, tiene como finalidad incluir el retardo de transferencia introducido por la capa de transporte USB, contemplando la influencia del parámetro ACL MTU. Sin embargo, su definición, que también ha de tener en cuenta la relación entre el tamaño de la trama L2CAP, (x) , y los parámetros ACL MTU (M_{ACL}) y el límite impuesto por la capa de transporte USB (M_{USB}), difiere de la dada para el modo EDR, calculándose según se indica a continuación:

$$C_{TX}^{USB}(x) = \begin{cases} \frac{1}{2}t_{USB}(x) & 0 \leq x \leq \min(M_{USB}, M_{ACL}) \\ \frac{1}{2}t_{USB}(M_{ACL}) & (x > M_{ACL}) \wedge (M_{ACL} \leq M_{USB}) \\ \frac{1}{2}t_{USB}(0) + \frac{1}{2}t_{USB}(x - M_{USB}) & (x > M_{USB}) \wedge (M_{ACL} > M_{USB}) \end{cases} \quad (5.15)$$

En la expresión anterior se diferencian tres casos:

- El tamaño de la trama L2CAP es inferior o igual al menor de los límites M_{ACL} y M_{USB} : Se entrega al *Host Controller* un único fragmento, de forma que tan solo se añade el retardo introducido por la capa de transporte para esa longitud: $\frac{t_{USB}(x)}{2}$.
- El tamaño de la trama L2CAP excede el mínimo de los límites M_{ACL} y M_{USB} : Se distinguen dos casos, en función de la relación entre ambos parámetros:
 - $M_{ACL} \leq M_{USB}$: En este caso se produce una fragmentación de acuerdo con el valor del parámetro ACL MTU, de forma que se computa el retardo consumido en la entrega al *Host Controller* del primer fragmento generado de tamaño M_{ACL} : $\frac{t_{USB}(M_{ACL})}{2}$.

- $M_{ACL} > M_{USB}$: Al ser superior M_{ACL} a M_{USB} , tienen lugar dos fragmentaciones sucesivas, la primera, según el valor de M_{ACL} y, la segunda, conforme al valor de M_{USB} . Por tanto, en el retardo introducido por la capa de transporte se distinguen dos contribuciones:
 1. Componente constante, $\frac{1}{2}t_{USB}(0)$, asociada al primer fragmento, cuya longitud es el mínimo de dos valores: ACL MTU (M_{ACL}) y 512 octetos (M_{USB}).
 2. Componente dependiente de la longitud, $\frac{1}{2}t_{USB}(x - \min(M_{ACL}, M_{USB}))$, correspondiente al residuo de $x - \min(M_{ACL}, M_{USB})$ octetos, que no excede los 512 octetos (M_{USB}).

5.3.5.3. Cálculo del retardo a nivel de RFCOMM

Para corregir la sobreestimación que tiene lugar al superarse los 1013 octetos ($L_R = 1008$ octetos), se ha redefinido la componente $t_{ACK}^c(x)$. Por tanto, el siguiente paso es reformular la ecuación que computa el retardo a nivel RFCOMM considerando el efecto de la transferencia por la capa de transporte USB como un término constante en dicha ecuación, tal y como se indica a continuación:

$$\bar{t}_R(x) = \begin{cases} t_{off} + t_{TX}^c(M(x)) & x \leq L_R \\ t_{off} + n_{fnf}(x) \cdot t_{ACK}^c(M(L_R)) + C_{fnf}^{USB} + & x > L_R \\ + t_{TX}^c(M(L_{ff}(x))) & \end{cases} \quad (5.16)$$

donde los términos $t_{TX}^c(x)$ y $t_{ACK}^c(x)$, se corresponden con los definidos por las ecuaciones 5.13 y 5.12 en el apartado 5.3.5.2.

En la ecuación 5.16, el término C_{fnf}^{USB} representa la corrección introducida para modelar el efecto de la capa de transporte USB en la transferencia de una trama *B-Frame* de L2CAP que encapsula una trama completa de RFCOMM (1008 octetos). Su definición se ve afectada

por el impacto adicional de la fragmentación de la trama L2CAP al tamaño dado por el parámetro ACL MTU. La expresión a la que se ha llegado, de forma heurística, es la siguiente:

$$C_{finf}^{USB} = \begin{cases} \frac{t_{USB} (\text{mín} (M_{ACL}, M_{USB}))}{2} & M_{ACL} \leq L_5 \\ \frac{t_{USB} (\text{mín} (M_{USB}, L_5)) + t_{USB}(0)}{2} & M_{ACL} > L_5 \end{cases} \quad (5.17)$$

Así, el *offset* añadido al retardo de transmisión del fragmento RFCOMM inicial (no único) depende de la relación entre los parámetros M_{USB} y M_{ACL} , según se describe a continuación:

- $M_{ACL} \leq L_5$: El comportamiento es diferente dependiendo de que el parámetro ACL MTU (M_{ACL}) configurado sea o no inferior al valor del parámetro M_{USB} :
 - $M_{ACL} < M_{USB}$: La trama de L2CAP es fragmentada al tamaño dado por el parámetro ACL MTU, generándose, al menos, un fragmento de tamaño M_{ACL} , inferior a M_{USB} , y sucediendo la siguiente secuencia:
 1. El primero de los fragmentos generados, de M_{ACL} octetos, se transfiere independientemente.
 2. Los siguientes fragmentos se almacenan temporalmente en el *HCI Driver*, resultando un bloque de longitud $(M(L_R) - M_{ACL})$ que, para ser la transferido por USB, es dividido conforme al tamaño definido por M_{USB} , generando un único fragmento de M_{USB} octetos y un residuo de longitud $(M(L_R) - M_{ACL} - M_{USB})$. Ambos fragmentos, ya en el *Host Controller*, son ensamblados en un bloque de $(M(L_R) - M_{ACL})$ octetos para su transmisión radio en uno o varios paquetes BB.

Por tanto, el tiempo de transferencia USB que se está computando en la primera rama de la ecuación 5.17, correspondería al tiempo consumido en la entrega al *Host Controller* del primer bloque de M_{ACL} octetos: $\frac{t_{USB}(M_{ACL})}{2}$.

- $M_{ACL} > M_{USB}$: La trama de L2CAP es fragmentada al tamaño dado por el parámetro ACL MTU, dando lugar a dos fragmentos, el primero de tamaño M_{ACL} , superior a M_{USB} , y el segundo de longitud $(M(L_R) - M_{ACL})$, de modo que:

1. El primero de los fragmentos generados, de M_{ACL} octetos, es fragmentado en bloques de 512 octetos (M_{USB}) para su transferencia a través de la capa de transporte, de forma que en el *Host Controller* son ensamblados en un bloque de M_{ACL} octetos para su transmisión radio en uno o varios paquetes BB.
2. El segundo de los fragmentos, de tamaño $(M(L_R) - M_{ACL})$, se transfiere independientemente.

Por consiguiente, en este caso, el retardo introducido por la capa de transporte USB, que se está computando en la primera rama de la ecuación 5.17, correspondería al tiempo consumido en la entrega al *Host Controller* del primer bloque de M_{USB} octetos: $\frac{t_{USB}(M_{USB})}{2}$.

- $M_{ACL} > L_5$: En este caso tiene lugar fragmentación en bloques de M_{USB} octetos para su envío a través de USB, bloques que son ensamblados en el *Host Controller* de forma que, para su transmisión vía radio, son fragmentados de acuerdo con la máxima capacidad de un paquete BB, dada por L_5 . Por otro lado, dado que en el extremo de recepción los datos se entregan al *Host* en bloques de L_5 octetos o de M_{USB} octetos, dependiendo de cuáles sea menor, el tiempo introducido por el transporte USB es la mitad de la suma de dos contribuciones:
 - Tiempo de transferencia USB de un bloque de L_5 octetos, o de M_{USB} octetos, es decir, de longitud dada por el menor de ambos parámetros, $t_{USB}(\min(M_{USB}, L_5))$.
 - Retardo mínimo asociado a la capa de transporte, $t_{USB}(0)$.

A modo de ejemplo, en la figura 5.29a se ha representado, para el modo EDR, cómo se ha supuesto que se lleva a cabo la transferencia de 1017 octetos en el caso de que el parámetro ACL MTU haya sido configurado a un valor superior a M_{USB} , concretamente de 680 octetos. Tal y como puede observarse, se generan en el *Host* dos fragmentos de 680 y 337 octetos, de forma que:

- El primero de los fragmentos (de 680 octetos), al exceder el valor de M_{USB} , para su transferencia por la interfaz USB, se divide en dos bloques, uno de 512 octetos y un

segundo de 168 octetos, bloques que son ensamblados al ser recibidos en el *Host Controller* para su envío a través de la interfaz radio en un solo paquete BB.

- Por su parte, el fragmento de 337 octetos, se envía directamente.

Es conveniente resaltar que el tiempo consumido en la entrega al *Host Controller* (por parte del *Host*) de este segundo fragmento queda enmascarado por el tiempo de transmisión radio del primer bloque de M_{ACL} octetos que, probablemente, tiene lugar en paralelo.

Adicionalmente, la figura 5.29b ilustra el comportamiento descrito para la transferencia de 1017 octetos en el caso de que el parámetro ACL MTU haya sido configurado a un valor inferior a M_{USB} , 384 octetos. Tal y como puede observarse, se generan tres fragmentos de 384, 384 y 249 octetos, de los cuales, el primero se envía directamente, mientras que el segundo y el tercero se recombinan (dando lugar a un bloque de 633 octetos) que, para su transferencia por la interfaz USB, son divididos, resultando dos fragmentos, uno de 512 octetos y un segundo de 121 octetos, los cuales son ensamblados para su transmisión vía radio en un único paquete BB.

Análogamente, a como se ha descrito para el caso en el que la ACL MTU toma el valor de 680 octetos, el tiempo empleado en la entrega al *Host Controller* (por parte del *Host*) del bloque de 633 octetos se compensa con el tiempo de transmisión radio del primer bloque de M_{ACL} octetos que, probablemente, tiene lugar en paralelo.

En definitiva, a partir del funcionamiento ilustrado en las figuras 5.28a, 5.28b, 5.29a y 5.29b, se puede concluir que:

- La fragmentación en bloques de 512 octetos, realizada para la transferencia USB, no influye en el esquema de transmisión, ya que afecta únicamente a la comunicación entre el *Host* y el *Host Controller*.
- La fragmentación que se lleva a cabo atendiendo al valor del parámetro ACL MTU (M_{ACL}) condiciona:
 - La transferencia a través de la capa de transporte.

- Los paquetes BB utilizados en la transmisión Bluetooth, de forma que si la longitud de los datos a transmitir es superior al valor del parámetro ACL MTU y esta, a su vez, es inferior a la máxima capacidad de un paquete de 5 *slots*, no se va a conseguir el máximo rendimiento porque se va a transmitir, un primer paquete con M_{ACL} octetos (ni siquiera completo) y un segundo paquete (o incluso más) con el número de octetos en los que la longitud de los datos de usuario excede el límite impuesto por el parámetro ACL MTU.

5.3.5.4. Presentación de los resultados

La validación de las correcciones introducidas en el modelo, se han llevado a cabo en cada uno de los modos de operación, para los valores del parámetro ACL MTU configurados por defecto en los dispositivos comerciales —384 y 310 octetos— y, también, para los valores siguientes:

- Modo EDR: 680, 512, 420 y 350 octetos, todos inferiores a la longitud de un paquete 3-DH5 (1021 octetos).
- Modo Básico:
 - Paquetes DH: 680 y 256 octetos.
 - Paquetes DM: 680 y 200 octetos.

Nótese que para el modo básico, la dependencia existente entre el valor del parámetro ACL MTU configurado y el límite inferior impuesto por la máxima capacidad de un paquete de 5 *slots* se comprueba en los siguientes casos:

- Paquetes DH: Con los valores de 310 y 256 octetos, que se encuentran por debajo de la máxima capacidad de un paquete DH5, 339 octetos.
- Paquetes DM: Con el valor de 200 octetos, menor que los 224 octetos que, como máximo pueden enviarse en un paquete DM5. Al respecto, es conveniente aclarar que no es viable fijar para el parámetro ACL MTU un valor inferior a los 200 octetos.

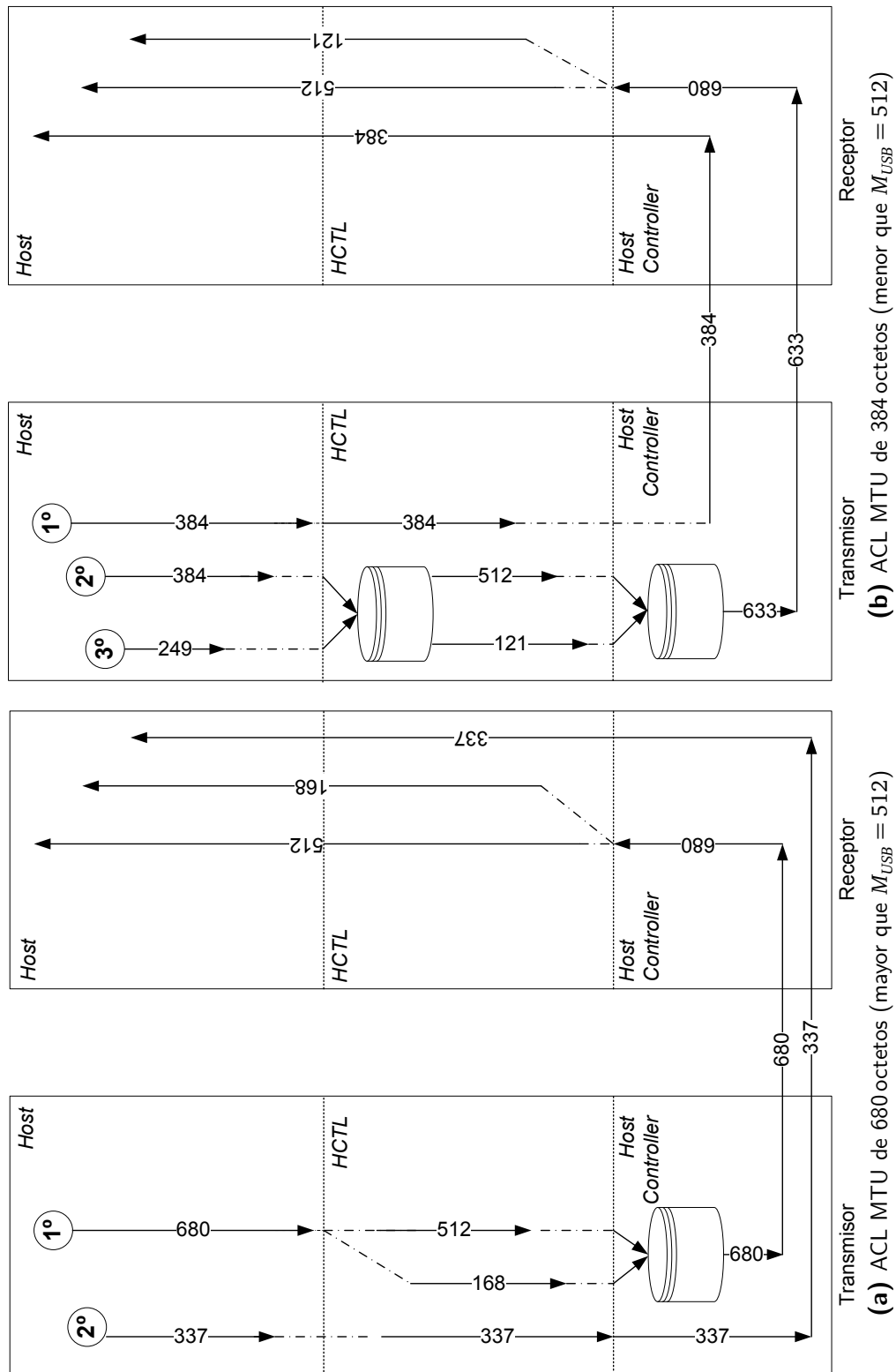


Figura 5.29.: Efecto de la fragmentación al tamaño del parámetro ACL MTU y de la capa de transporte para la transferencia de una trama L2CAP que encapsula una trama de RFCOMM de longitud máxima (1008 octetos de datos de usuario).

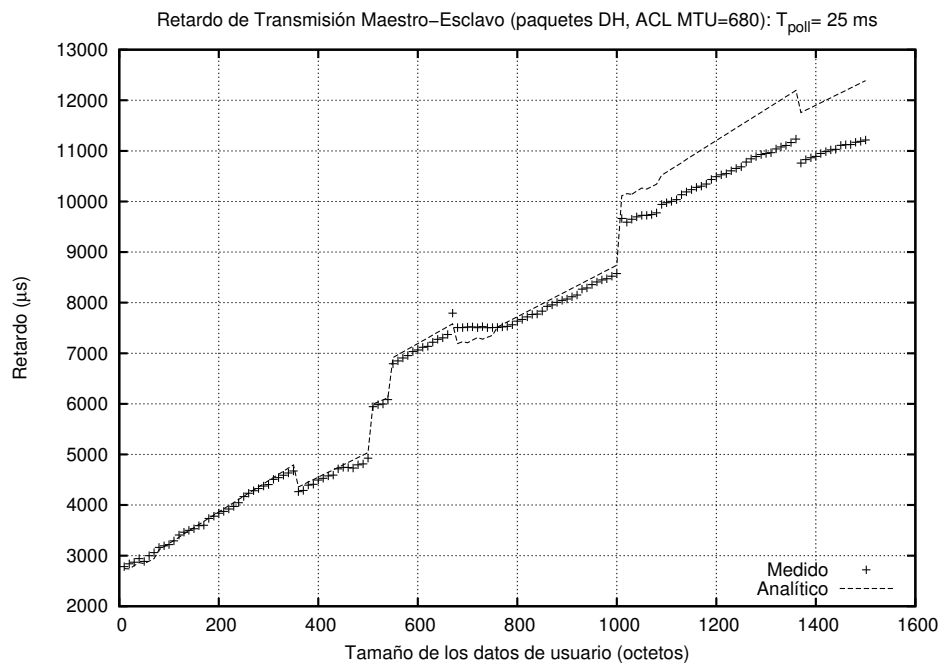
A continuación, en los siguientes epígrafes, se presentan los resultados de las medidas realizadas, en primer lugar para el modo EDR y después para el modo básico.

Paquetes DH en el modo EDR

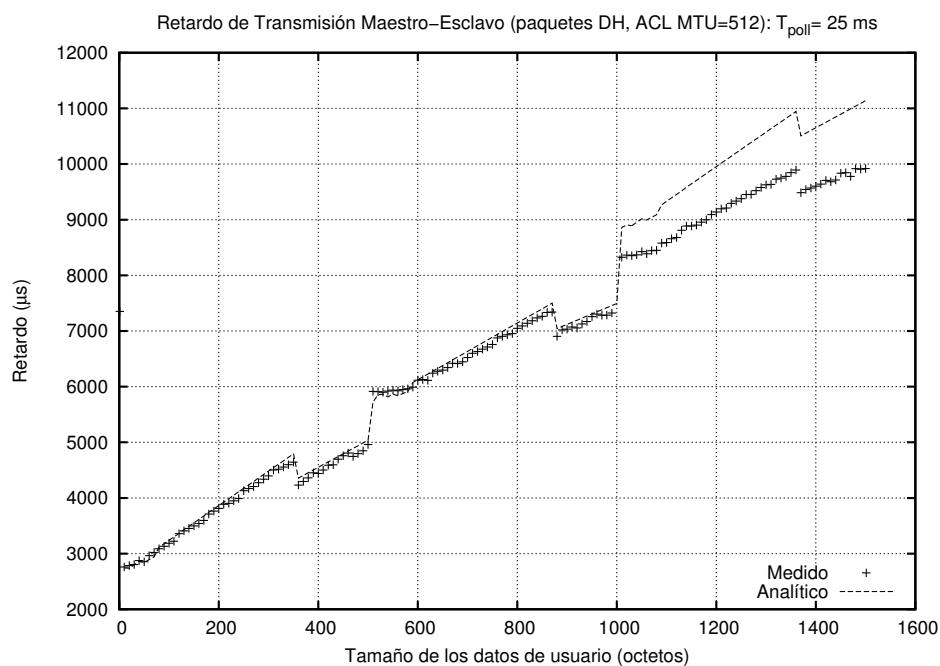
El retardo calculado teóricamente a partir de la ecuación 5.16, en comparación con los retardos medios medidos, sin calidad de servicio ($T_{poll}=25$ ms), cuando el parámetro ACL MTU se configura a los valores de 680, 512, 420, 384, 350 y 310 octetos en el rango de longitudes comprendido entre 10 y 1500 octetos, se representa en las gráficas de las figuras 5.30a, 5.30b, 5.31a, 5.31b, 5.32a y 5.32b. En todos los casos puede observarse que se consigue modelar el efecto de la fragmentación de acuerdo con los parámetros ACL MTU (M_{ACL}) y el límite impuesto por la capa de transporte USB (M_{USB}), siempre que no se exceda el tamaño de la MTU de L2CAP para RFCOMM, $M_R = 1013$ octetos ($L_R = 1008$ octetos), inferior a la capacidad de un único paquete de 5 slots de máxima capacidad: 3-DH5.

Nótese que, en los casos en los que el parámetro ACL MTU toma los valores de 512 y 680 octetos, pese a que se produce una sobrestimación, las curvas del retardo teórico y medido coinciden, diferenciándose únicamente en un valor constante y en una mínima desviación en las pendientes. Sin embargo, en lo que concierne a los casos en los que el parámetro ACL MTU es inferior a 512 octetos, $M_{ACL} < M_{USB}$, cabe resaltar que el retardo obtenido para longitudes de datos de usuario superiores a los 1008 octetos no se ve afectado por la fragmentación al tamaño del parámetro ACL MTU. Esto es, puede observarse que, a partir de este punto, el retardo previsto teóricamente responde al mismo patrón inicial, de forma que al alcanzar los $1008 + M_{ACL}$ octetos, se produce un escalón ascendente (equivalente al que tiene lugar en la longitud M_{ACL} octetos). Por contra, en el retardo obtenido empíricamente no se reproduce el comportamiento descrito. Por consiguiente, puede concluirse que el efecto del parámetro ACL MTU no tiene el mismo impacto sobre el cálculo del retardo de transmisión para el primer (y único) fragmento RFCOMM, que para un fragmento final en el que se encapsulan el número de octetos en el que se excede la máxima capacidad de una trama RFCOMM ($L_R = 1008$ octetos).

Posteriormente, en el apartado 5.3.6, teniendo en cuenta estas consideraciones, se propone una corrección que, aunque tampoco se ajusta completamente a la realidad en todos los casos, proporciona una mejor aproximación.

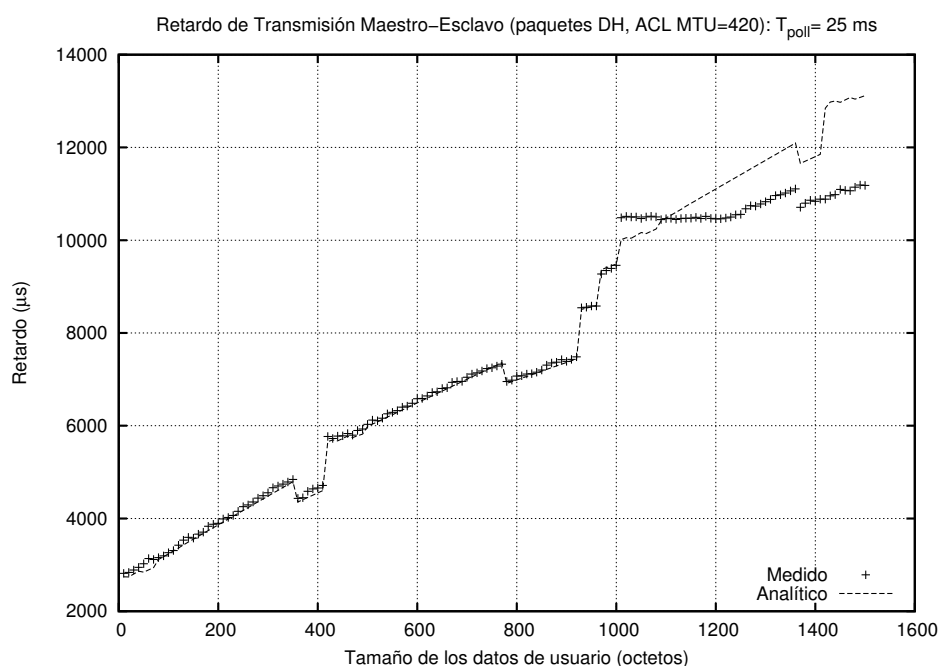


(a) ACL MTU de 680 octetos

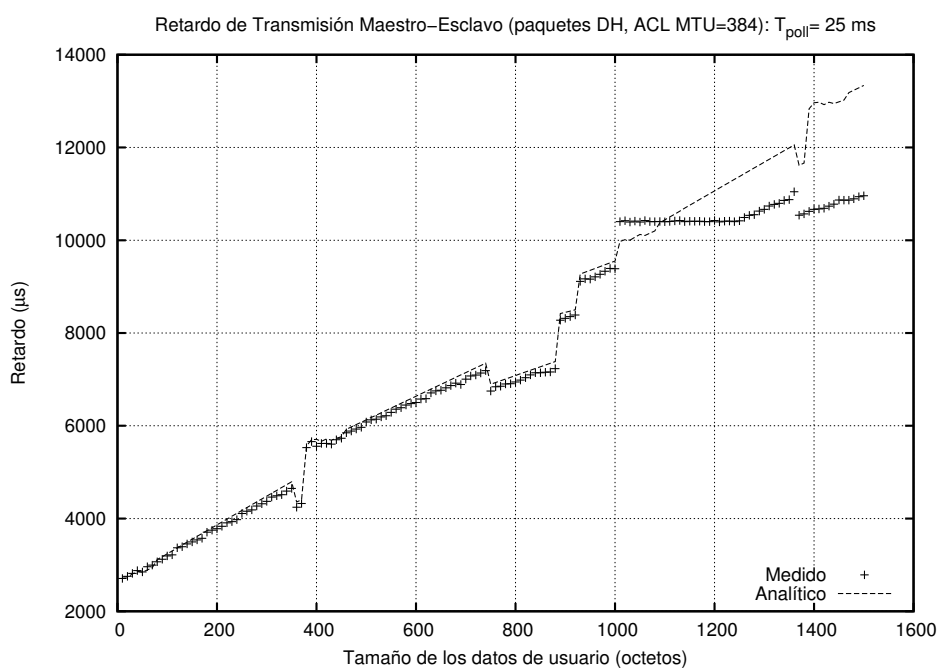


(b) ACL MTU de 512 octetos

Figura 5.30.: Comparación del retardo medido y el teórico incluyendo la suma de los efectos de la fragmentación según el parámetro ACL MTU y de la capa de transporte USB, en el sentido maestro-esclavo, para paquetes DH del modo EDR y valores del parámetro ACL MTU de 680 y 512 octetos (Revisión de HCI: 0xc5c).

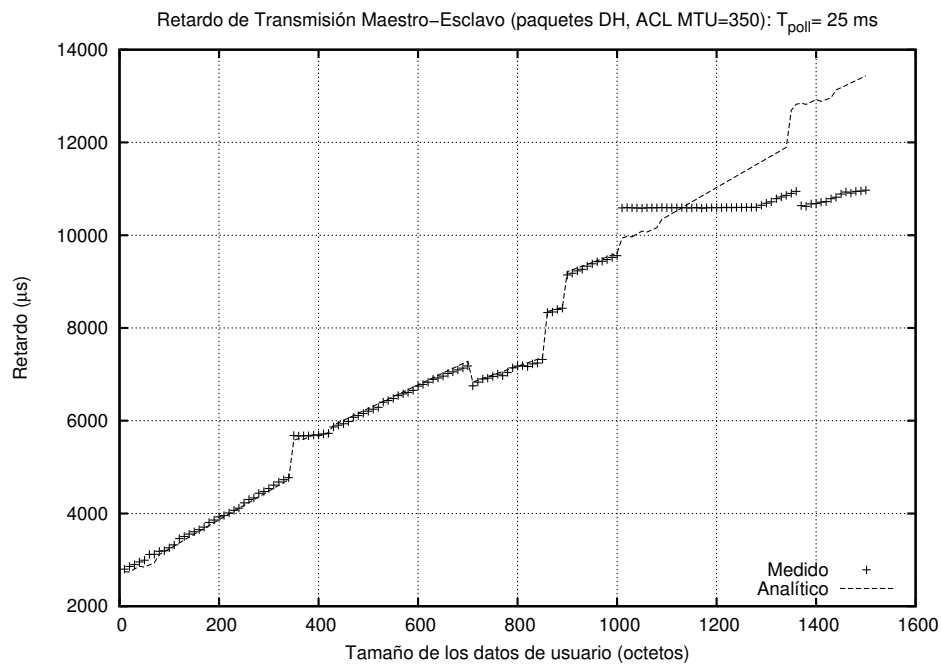


(a) ACL MTU de 420 octetos

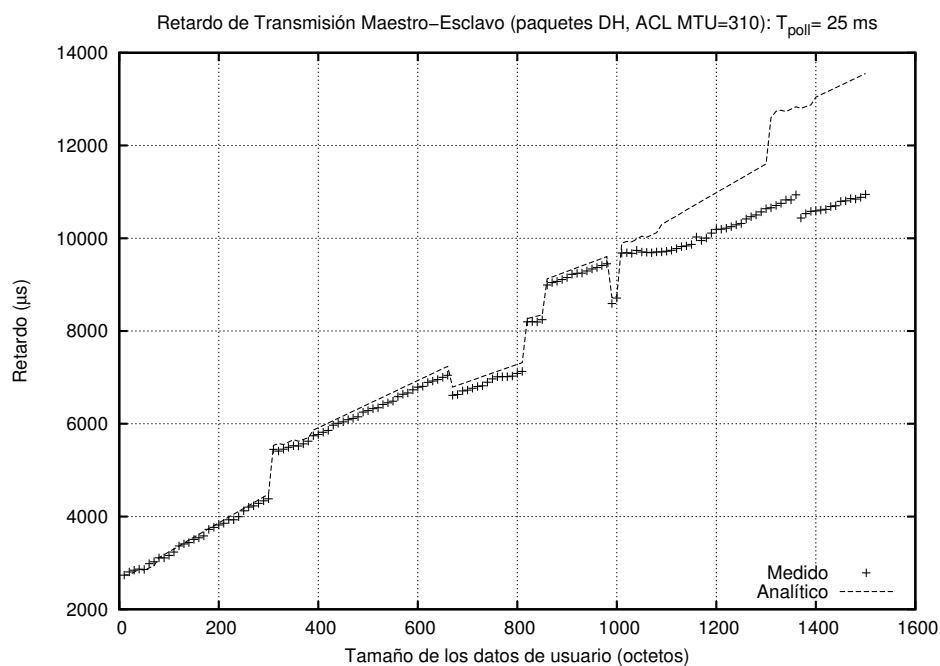


(b) ACL MTU de 384 octetos

Figura 5.31.: Comparación del retardo medido y el teórico incluyendo la suma de los efectos de la fragmentación según el parámetro ACL MTU y de la capa de transporte USB, en el sentido maestro-esclavo, para paquetes DH del modo EDR y valores del parámetro ACL MTU de 420 y 384 octetos (Revisión de HCI: 0xc5c)



(a) ACL MTU de 350 octetos



(b) ACL MTU de 310 octetos

Figura 5.32.: Comparación del retardo medido y el teórico incluyendo la suma de los efectos de la fragmentación según el parámetro ACL MTU y de la capa de transporte USB, en el sentido maestro-esclavo, para paquetes DH del modo EDR y valores del parámetro ACL MTU de 350 y 310 octetos (Revisión de HCI: 0xc5c)

Paquetes DM y DH en el modo básico

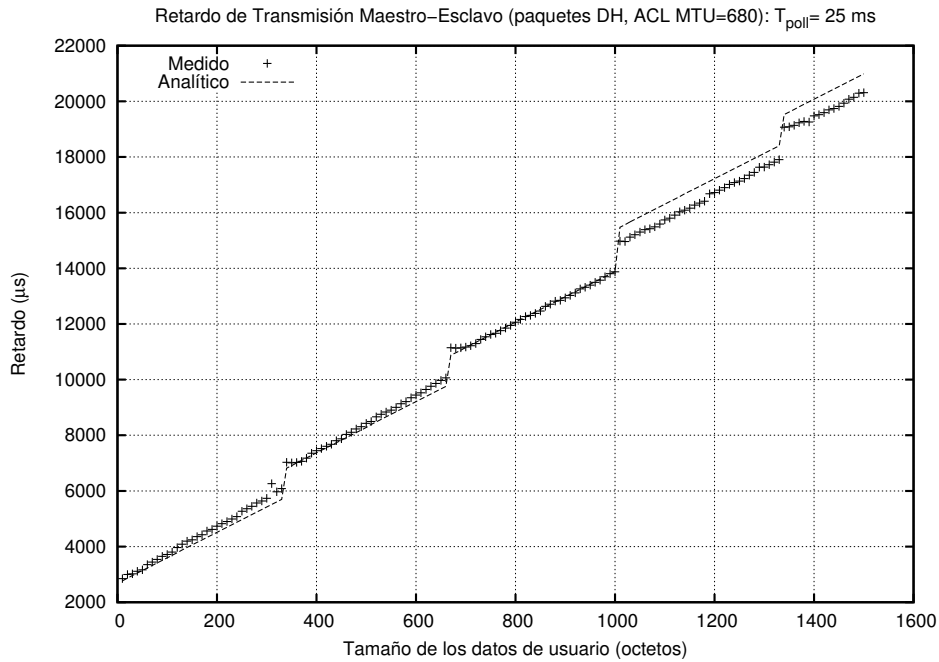
Con el objetivo de comprobar la validez de la modificación introducida en el modelo para incluir el efecto de la fragmentación de acuerdo con los parámetros ACL MTU (M_{ACL}) y el límite impuesto por la capa de transporte USB (M_{USB}) se han llevado a cabo medidas, utilizando paquetes DH del modo básico, para los valores del parámetro ACL MTU configurados por defecto en los dispositivos comerciales —384 y 310 octetos— y también para 680 y 256 octetos. Adicionalmente, para los paquetes DM, dado que los valores 384, 310 y 256 octetos son superiores a la máxima capacidad de un paquete DM5, 224 octetos, ha sido necesario realizar una prueba adicional reduciendo el valor del parámetro ACL MTU a 200 octetos, a fin de poder constatar la relación entre dicho parámetro y la máxima capacidad de un paquete de 5 *slots*.

En las gráficas de las figuras 5.33a, 5.33b, 5.34a, 5.34b se ha representado el retardo calculado teóricamente a partir de la ecuación 5.16, en comparación con los retardos medios medidos, sin calidad de servicio ($T_{poll}=25$ ms), utilizando paquetes DH, cuando el parámetro ACL MTU se configura a los valores de 680, 384, 310 y 256 octetos en el rango de longitudes comprendido entre 10 y 1500 octetos. Por otro lado, en las figuras 5.35a y 5.35b se incluyen los resultados obtenidos al emplear paquetes DM y fijando un valor de 680 y 200 octetos para el parámetro ACL MTU.

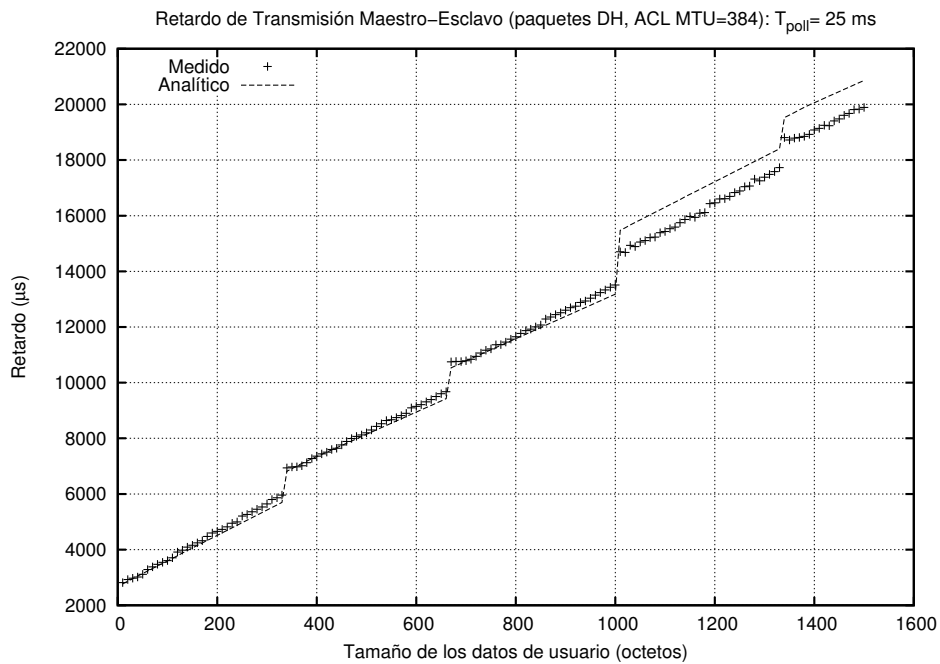
Nótese que los casos en los que el parámetro en cuestión, M_{ACL} , toma los valores de 256 y 310 octetos, inferiores a la máxima capacidad de un paquete DH5, el modelo teórico responde al comportamiento real, es decir, puede apreciarse que el primer incremento abrupto tiene lugar exactamente en el límite dado por el parámetro M_{ACL} , 256 y 310 octetos y no en los 339 octetos como sucede cuando el valor del parámetro ACL MTU es configurado por encima del máximo número de octetos que pueden enviarse en un paquete de 5 *slots* como en el caso en que dicho parámetro vale 384 octetos ($M_{ACL} = 384$ octetos). Análogamente, para los paquetes DM, cuando el parámetro M_{ACL} se configura a 200 octetos, valor que se encuentra por debajo de la máxima capacidad de un paquete DM5, el primer incremento abrupto se produce al alcanzar esta longitud, en vez de a los 224 octetos, tal y como se observa en la figura 5.35b. Sin embargo, tanto para paquetes DH como DM, este comportamiento no se reproduce cuando se excede el tamaño de la MTU de L2CAP para RFCOMM, $M_R = 1013$ octetos ($L_R = 1008$ octetos), límite a partir del cual se sobreestima el retardo

medido para todos los casos. Adicionalmente se ha constatado, tanto para paquetes DH como DM, que el comportamiento es el mismo en el caso límite ($M_{USB} = M_{ACL} = 512$ octetos), según se ilustra en las gráficas de la figura 5.36a y 5.36b.

Por consiguiente, tal y como sucede para el modo EDR, se concluye que el efecto de la capa de transporte no tiene el mismo impacto sobre el cálculo del retardo de transmisión para el primer (y único) fragmento RFCOMM, que para un fragmento final que transporta el número de octetos en el que se excede la máxima capacidad de una trama RFCOMM ($L_R = 1008$ octetos).

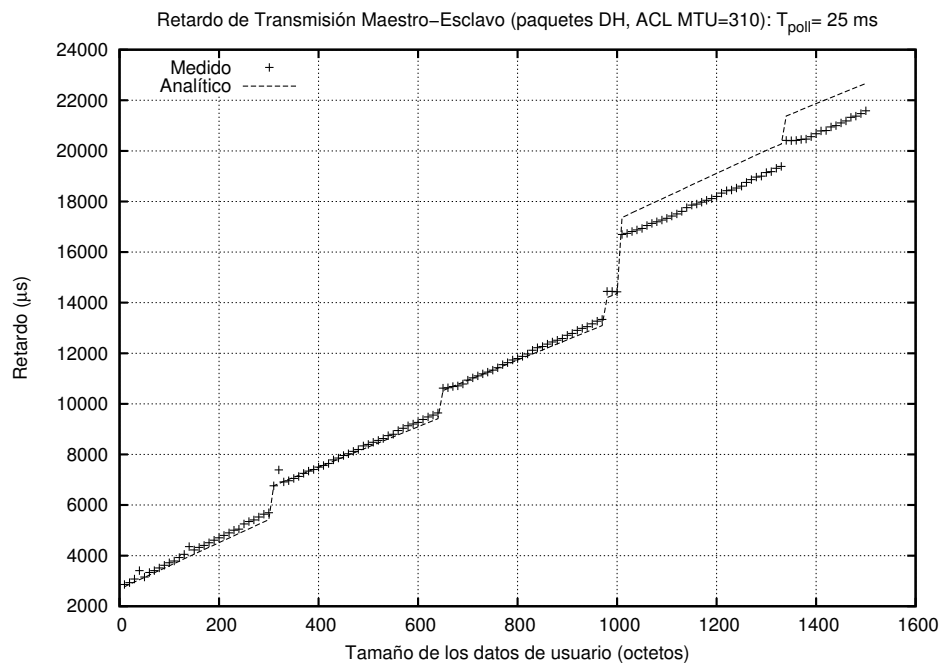


(a) ACL MTU de 680 octetos (superior a $M_{USB} = 512$ octetos y a $L_5 = 339$ octetos)

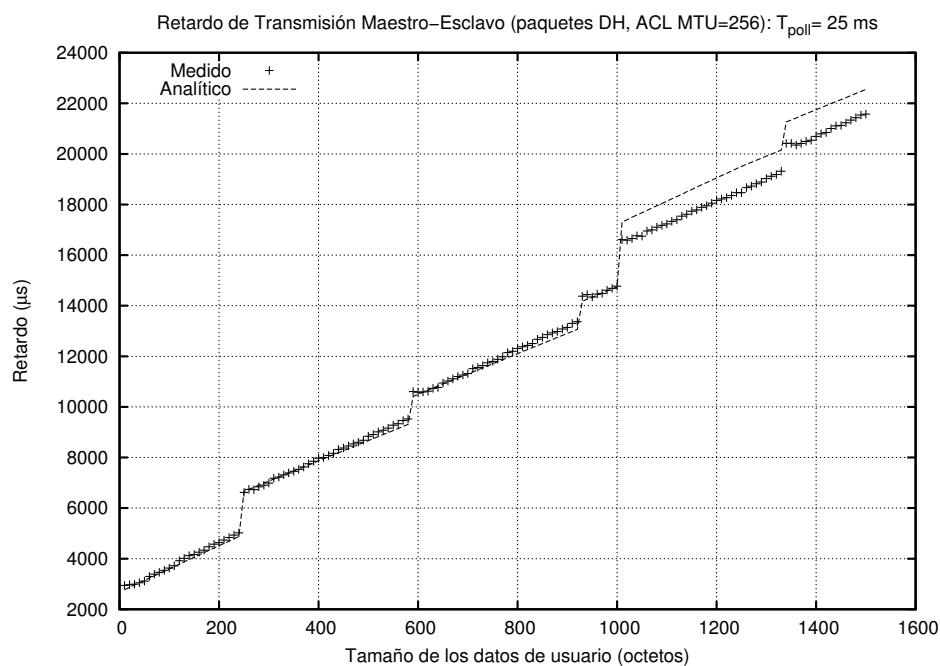


(b) ACL MTU de 384 octetos (inferior a $M_{USB} = 512$ octetos y superior a $L_5 = 339$ octetos)

Figura 5.33.: Comparación del retardo medido y el teórico incluyendo la suma de los efectos de la fragmentación según el parámetro ACL MTU y de la capa de transporte USB, en el sentido maestro-esclavo, para paquetes DH del modo básico y valores del parámetro ACL MTU de 680 y 384 octetos (Revisión de HCI: 0xc5c)

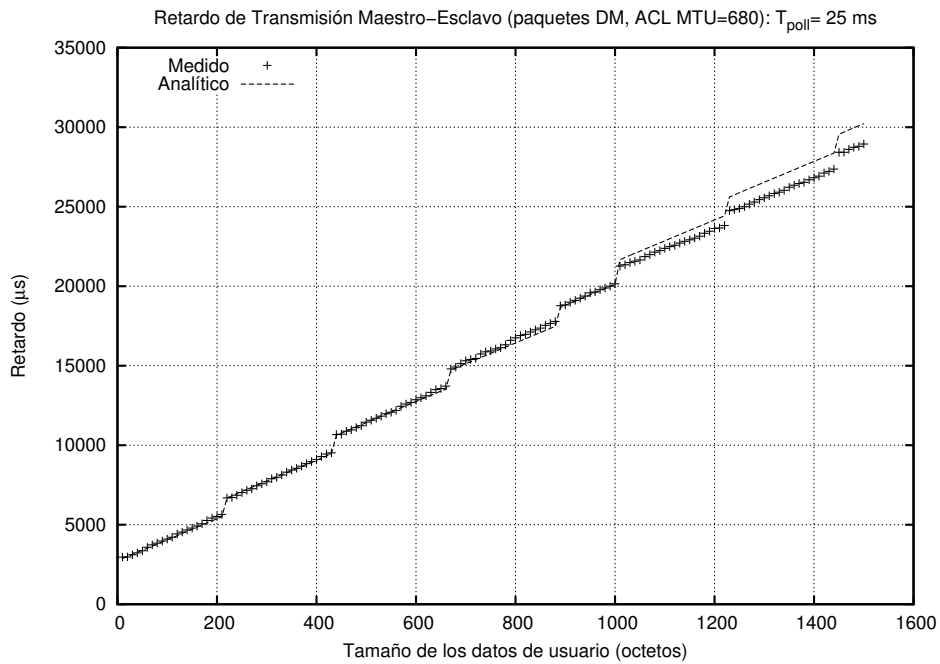


(a) ACL MTU de 310 octetos (inferior a $M_{USB} = 512$ octetos y a $L_5 = 339$ octetos)

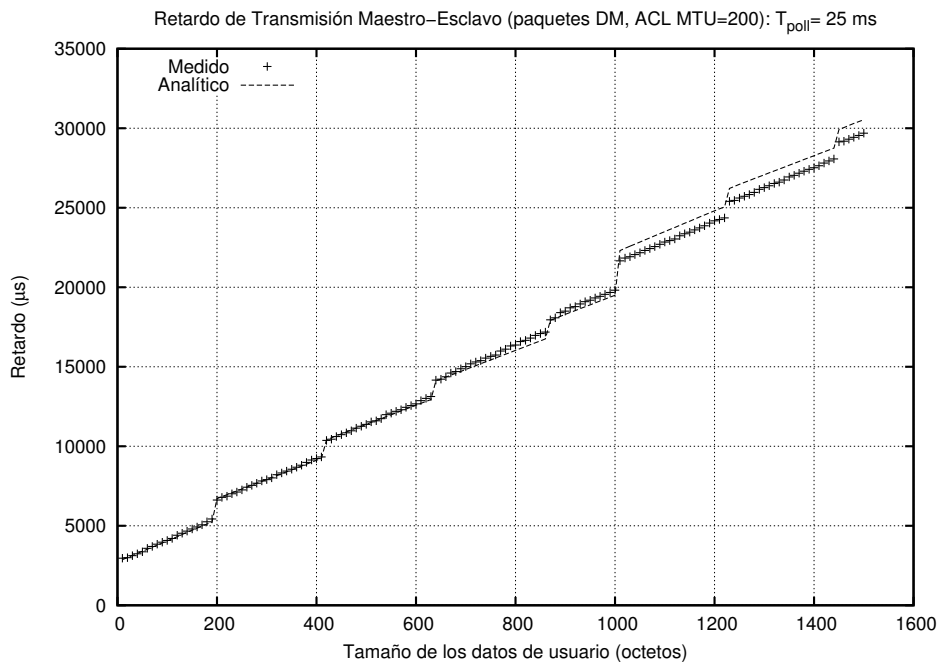


(b) ACL MTU de 256 octetos (inferior a $M_{USB} = 512$ octetos y a $L_5 = 339$ octetos)

Figura 5.34.: Comparación del retardo medido y el teórico incluyendo la suma de los efectos de la fragmentación según el parámetro ACL MTU y de la capa de transporte USB, en el sentido maestro-esclavo, para paquetes DH del modo básico y valores del parámetro ACL MTU de 310 y 256 octetos (Revisión de HCI: 0xc5c)

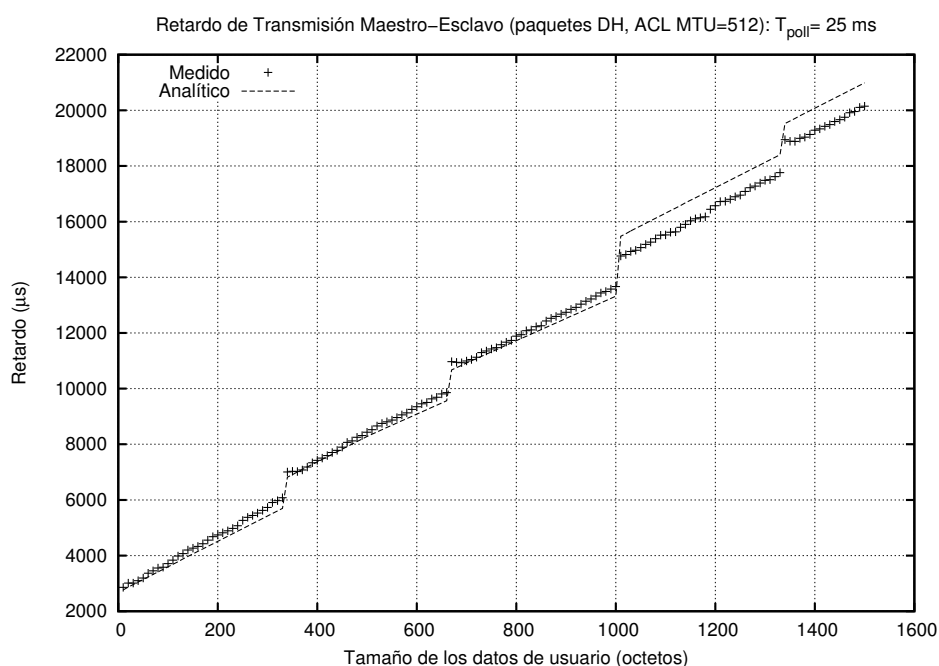


(a) ACL MTU de 680 octetos (superior a $M_{USB} = 512$ octetos y a $L_5 = 224$ octetos)

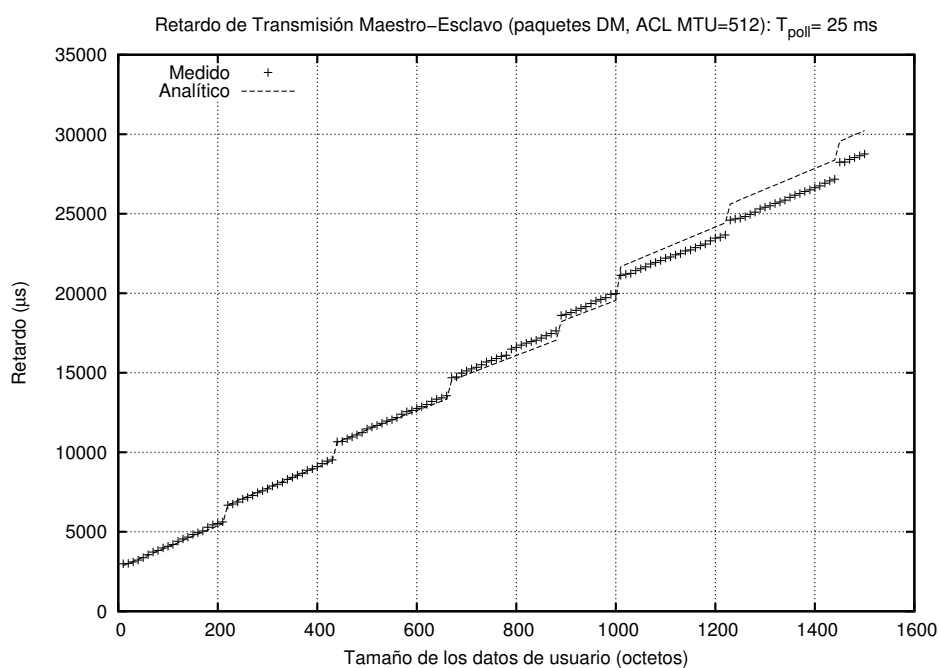


(b) ACL MTU de 200 octetos (inferior a $M_{USB} = 512$ octetos y a $L_5 = 224$ octetos)

Figura 5.35.: Comparación del retardo medido y el teórico incluyendo la suma de los efectos de la fragmentación según el parámetro ACL MTU y de la capa de transporte USB, en el sentido maestro-esclavo, para paquetes DM y valores del parámetro ACL MTU de 680 y 200 octetos (Revisión de HCI: 0xc5c)



(a) Paquetes DH ($M_{ACL} = M_{USB} = 512$ octetos y $L_5 = 339$ octetos)



(b) Paquetes DM ($M_{ACL} = M_{USB} = 512$ octetos y $L_5 = 224$ octetos)

Figura 5.36.: Comparación del retardo medido y el teórico incluyendo la suma de los efectos de la fragmentación según el parámetro ACL MTU y de la capa de transporte USB, en el sentido maestro-esclavo, para paquetes DH y DM del modo básico y un valor del parámetro ACL MTU de 512 octetos (Revisión de HCI: 0xc5c)

5.3.6. Modificación del modelo teórico para contemplar el efecto de la capa de transporte para tamaños de datos de usuario superiores a 1008 octetos

En esta sección se plantea cómo refinar el modelo teórico para evitar la sobreestimación que tiene lugar al excederse el tamaño de la MTU de L2CAP para RFCOMM, $M_R = 1013$ octetos ($L_R = 1008$ octetos), tanto para el modo básico y como para el modo EDR.

La modificación que se va a introducir en el modelo para el cálculo del retardo a nivel RFCOMM se basa en la observación de que para determinar el retardo de transmisión asociado a un fragmento intermedio (segundo, en los escenarios de prueba contemplados, para los que la longitud máxima de los datos de usuario es de 1500 octetos), ha de tener en cuenta que:

1. El retardo introducido por la transferencia USB no tiene el mismo impacto que en un fragmento inicial.
2. No se ve afectado por la fragmentación al tamaño del parámetro ACL MTU, tal y como se ha observado a partir de las medidas realizadas, presentadas en el apartado 5.3.5.

Por consiguiente, la corrección a realizar se limita a la segunda rama de la ecuación que computa el retardo a nivel de RFCOMM para datos de usuario cuya longitud excede los 1008 octetos (L_R), que pueden transmitirse en una única trama RFCOMM. Así, la ecuación 5.16 ha de redefinirse del siguiente modo:

$$\bar{t}_R(x) = \begin{cases} t_{off} + t_{TX}^c(M(x)) & x \leq L_R \\ t_{off} + n_{fnf}(x) \cdot t_{ACK}^c(M(L_R)) + C_{fnf}^{USB} + \\ + t_{TX}(M(L_{ff}(x))) + C_{ff}^{USB}(M(L_{ff}(x))) & x > L_R \end{cases} \quad (5.18)$$

Nótese que, respecto a la ecuación 5.16, la modificación introducida ha consistido únicamente en reemplazar, en la segunda rama que computa el retardo en los casos en los que los datos de usuario son superiores a 1008 octetos (L_R), la componente $t_{TX}^c(M(L_{ff}(x)))$ por la suma de la componente de retardo básica, $t_{TX}(x)$, y del término $C_{ff}^{USB}(x)$, que

representa el retardo introducido por la capa de transporte para una trama RFCOMM final, cuya definición difiere según el modo de operación:

- Modo básico: El retardo de transmisión Bluetooth de la segunda trama RFCOMM no se ve incrementado por el tiempo consumido en la transferencia USB. Por tanto, el término $C_{ff}^{USB}(x)$ es nulo:

$$C_{ff}^{USB}(x) = 0 \quad (5.19)$$

- Mode EDR: Al retardo de transmisión Bluetooth, $t_{TX}(x)$, se suma la mitad del retardo introducido por la capa de transporte, de forma que el término $C_{ff}^{USB}(x)$ se define como:

$$C_{ff}^{USB}(x) = \frac{t_{USB}(x)}{2} \quad (5.20)$$

Así, en la ecuación 5.18, la primera rama computa el retardo de transmisión, $t_{TX}^c(M(x))$, para datos de usuario hasta un máximo de $L_R = 1008$ octetos, utilizando la ecuación 5.13. Sin embargo, cuando este límite es excedido, el tiempo total se obtiene mediante la segunda rama, de modo que:

- Al tiempo de transmisión del fragmento inicial, incluyendo el *slot* de confirmación, $t_{ACK}^c(x)$, computado a partir de la ecuación 5.9, que tiene en cuenta el efecto de la fragmentación según el valor del parámetro ACL MTU configurado, se suma el factor de corrección C_{ff}^{USB} , dado por la ecuación 5.17 para contemplar la influencia de la capa de transporte.
- El tiempo de transmisión del fragmento intermedio (segundo) es el retardo dado por la componente básica, $t_{TX}(x)$, al que sólo en el modo EDR se añade la mitad del retardo introducido por la capa de transporte.

A diferencia del enfoque seguido en las secciones 5.3.4 y 5.3.5, se comienza exponiendo los resultados de las medidas realizadas para validar esta última corrección para el modo básico en el apartado 5.3.6.1, para proseguir analizando el comportamiento en el modo EDR, de mayor complejidad, en el apartado 5.3.6.2.

5.3.6.1. Paquetes DM y DH en el modo básico

El resultado de la modificación introducida para corregir la sobrestimación que tiene lugar cuando el tamaño de los datos de usuario tienen que transmitirse en más de una trama RFCOMM se ilustra, para el modo básico, en las figuras 5.37a, 5.37b, 5.38a y 5.38b en las que se compara, para los paquetes DH del modo básico, el retardo calculado de acuerdo con la ecuación 5.18, con las medidas realizadas con dispositivos Bluetooth 2.0 + EDR y versión de *firmware* 0xc5c, en el sentido de transmisión maestro-esclavo y sin haber configurado la calidad de servicio ($T_{poll} = 25 \text{ ms}$). Puede comprobarse que, con la corrección introducida, desaparece la sobreestimación a partir de los 1008 octetos de datos de usuario. Asimismo se ha constatado la validez del modelo cuando se utilizan paquetes DM, habiendo establecido un valor de 25 ms para el intervalo de *poll* (T_{poll}), según se muestra en las figuras 5.39a y 5.39b. Concretamente, las curvas representadas corresponden a los casos en los que el parámetro ACL MTU se ha configurado a 680 (figura 5.39a) y 200 octetos (figura 5.39b), valores que se encuentran, respectivamente, por encima y por debajo de la máxima capacidad de un paquete DM5 (224 octetos). También, se ha verificado, tanto para paquetes DH como DM, que el comportamiento es el mismo en el caso límite ($M_{USB} = M_{ACL} = 512 \text{ octetos}$), según se ilustra en las gráficas de la figura 5.40a y 5.40b.

Adicionalmente, con el objeto de ratificar la validez del modelo refinado en el presente apartado, en las figuras 5.41a y 5.41b se muestra el resultado satisfactorio del ajuste realizado para dispositivos Bluetooth 2.0 + EDR de la versión de *firmware* 0x7a6, habiendo fijado un intervalo de *poll* de 10 ms y 25 ms, aunque se recuerda que en este caso no existe dependencia del retardo respecto del valor del parámetro ACL MTU configurado.

Para finalizar, en las figuras 5.42a y 5.42b se representa, a modo de ejemplo, el retardo medido con dispositivos Bluetooth 2.0 + EDR de la versión de *firmware* 0xc5c, en comparación con el calculado teóricamente, en el sentido de transmisión esclavo-maestro, para valores del intervalo de *poll* de 10 ms y 20 ms, pese a que en este caso el comportamiento también es independiente del parámetro ACL MTU utilizado.

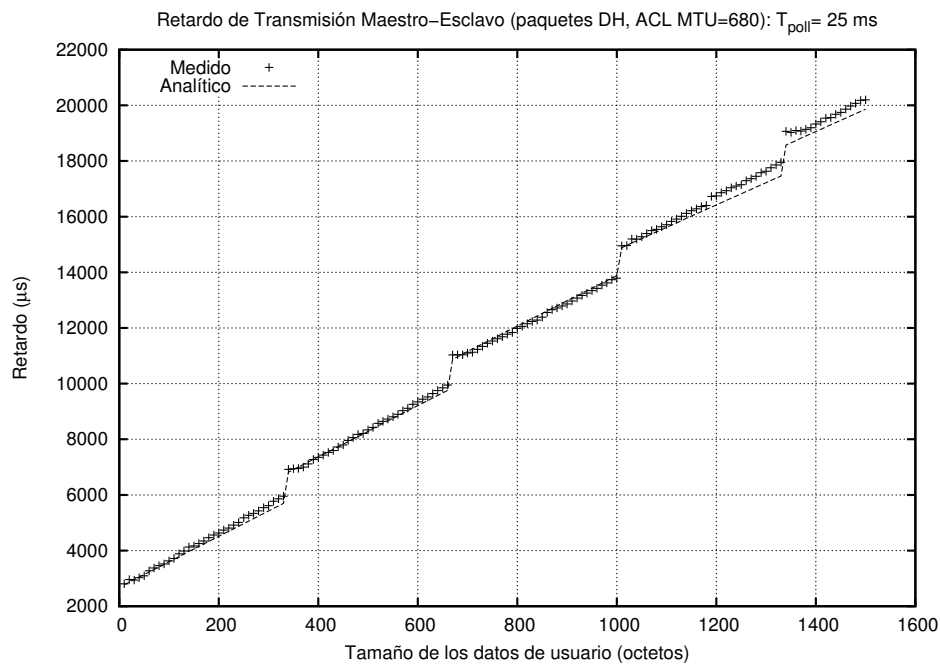
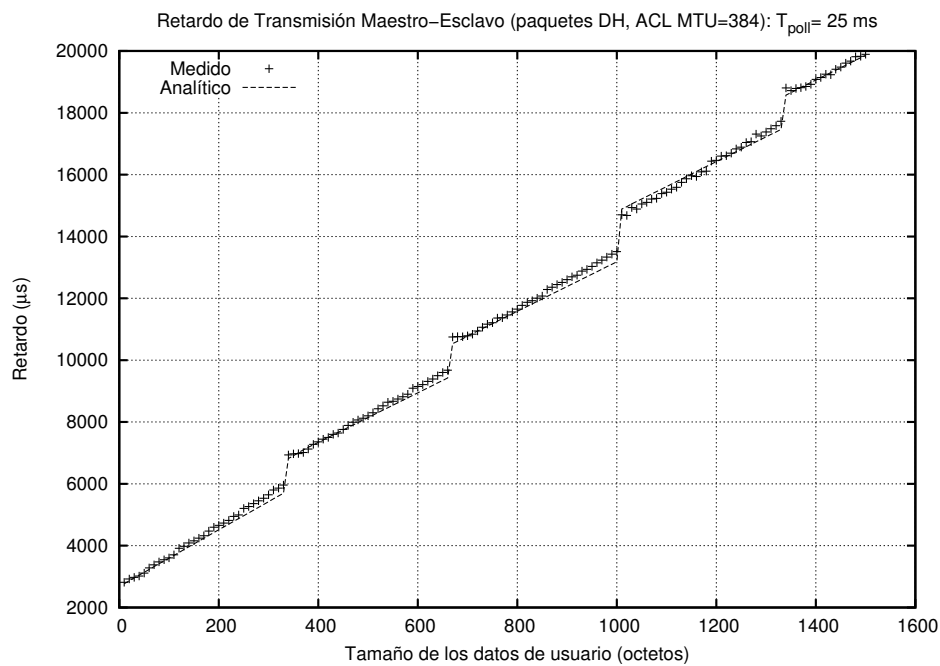
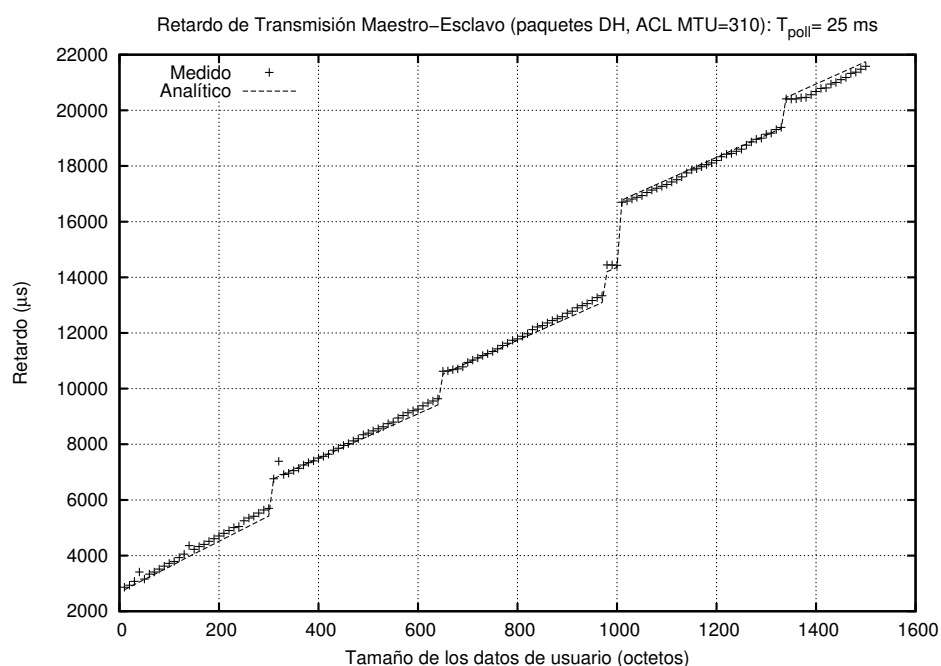
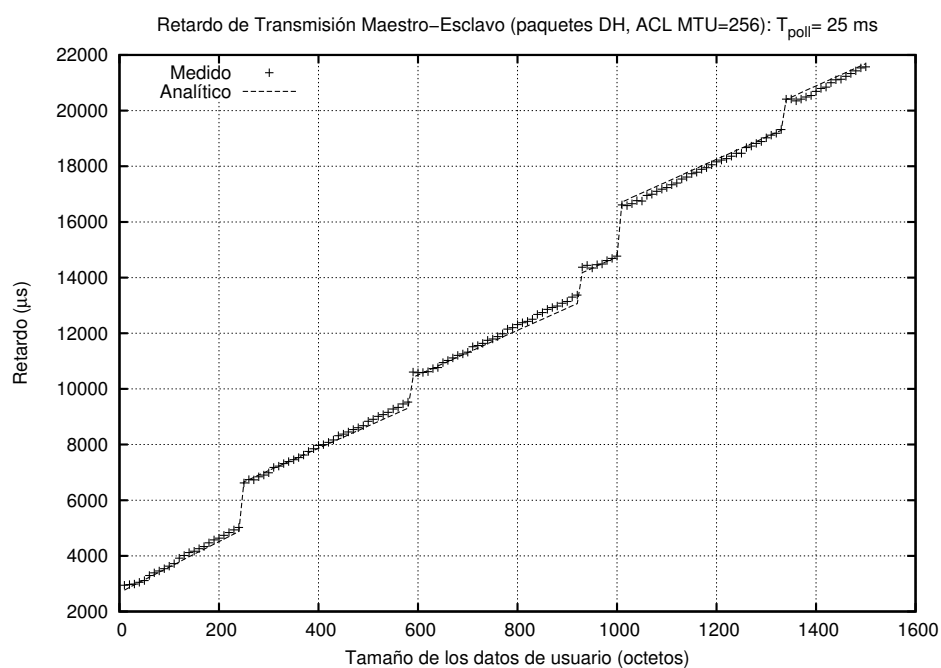
(a) ACL MTU de 680 octetos, $T_{poll}=25$ ms(b) ACL MTU de 384 octetos, $T_{poll}=25$ ms

Figura 5.37.: Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DH básicos: DH1, DH3 y DH5 y valores del parámetro ACL MTU de 680 y 384 octetos (Revisión de HCI: 0xc5c).



(a) ACL MTU de 310 octetos, $T_{poll}=25$ ms



(b) ACL MTU de 256 octetos, $T_{poll}=25$ ms

Figura 5.38.: Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DH básicos: DH1, DH3 y DH5 y valores del parámetro ACL MTU de 310 y 256 octetos (Revisión de HCI: 0xc5c)

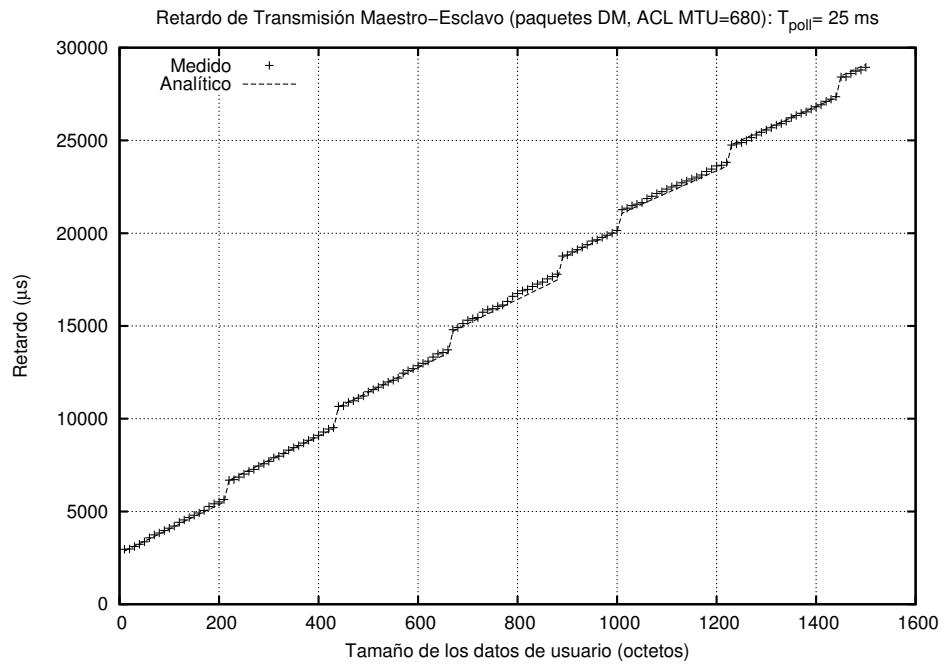
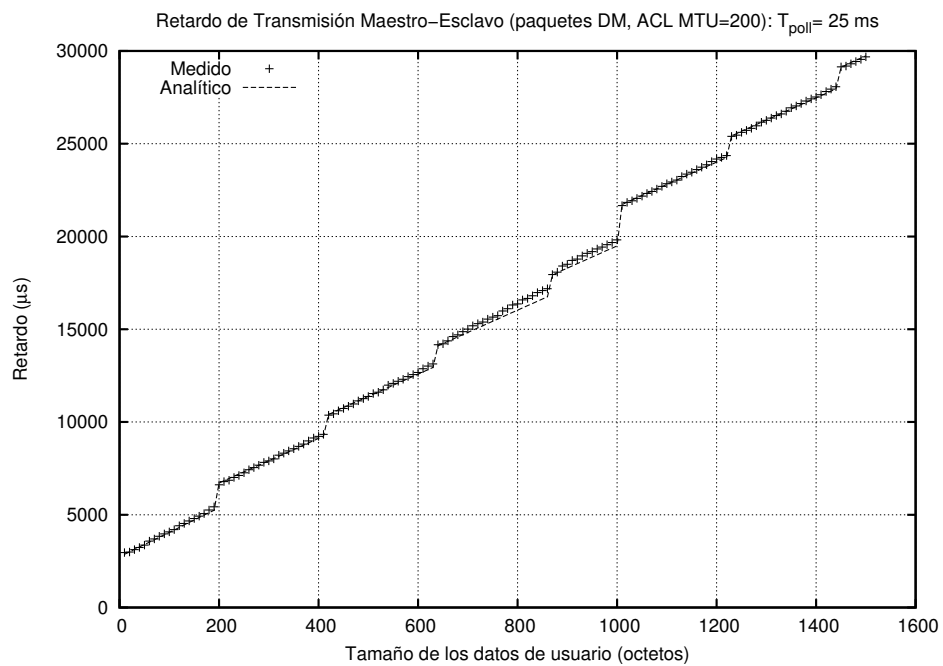
(a) ACL MTU de 680 octetos, $T_{poll}=25$ ms(b) ACL MTU de 200 octetos, $T_{poll}=25$ ms

Figura 5.39.: Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DM y valores del parámetro ACL MTU de 680 y 200 octetos (Revisión de HCI: 0xc5c).

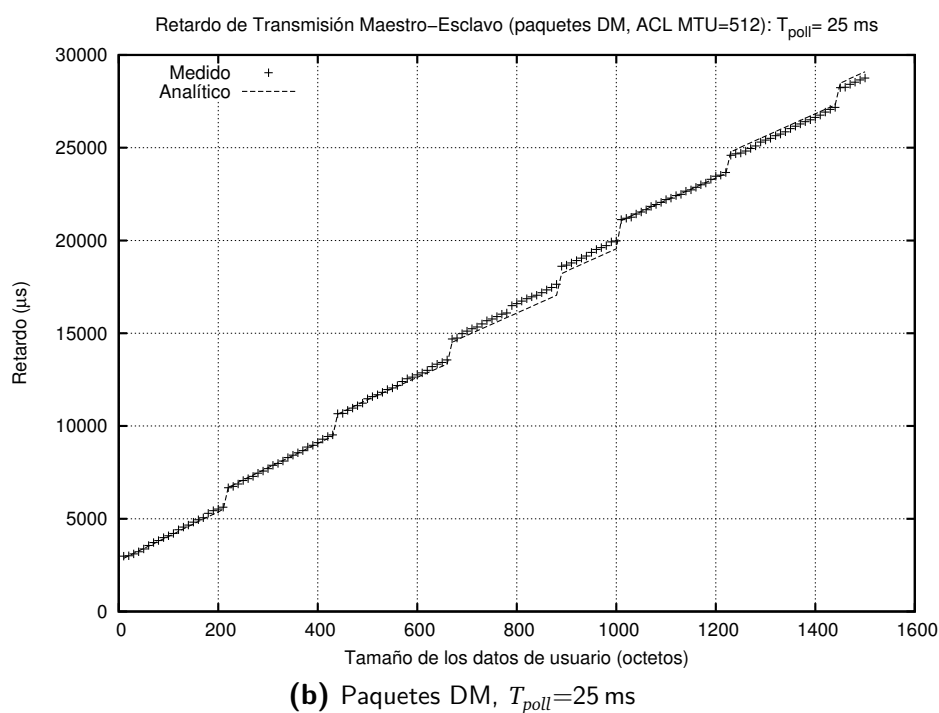
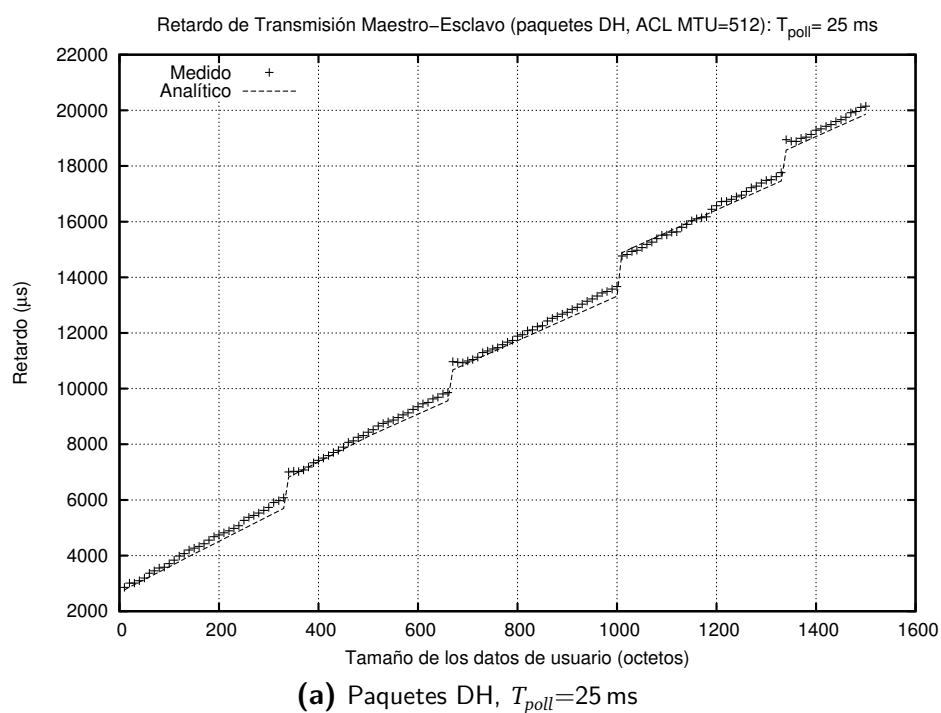


Figura 5.40.: Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DM y DH básicos, y un valor del parámetro ACL MTU de 512 octetos (Revisión de HCI: 0xc5c).

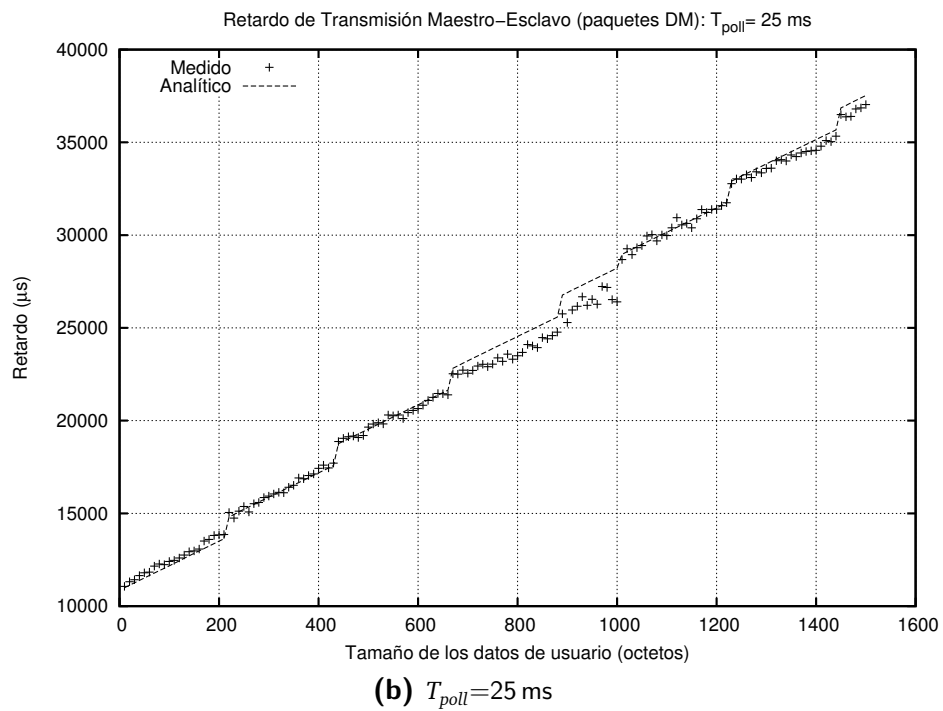
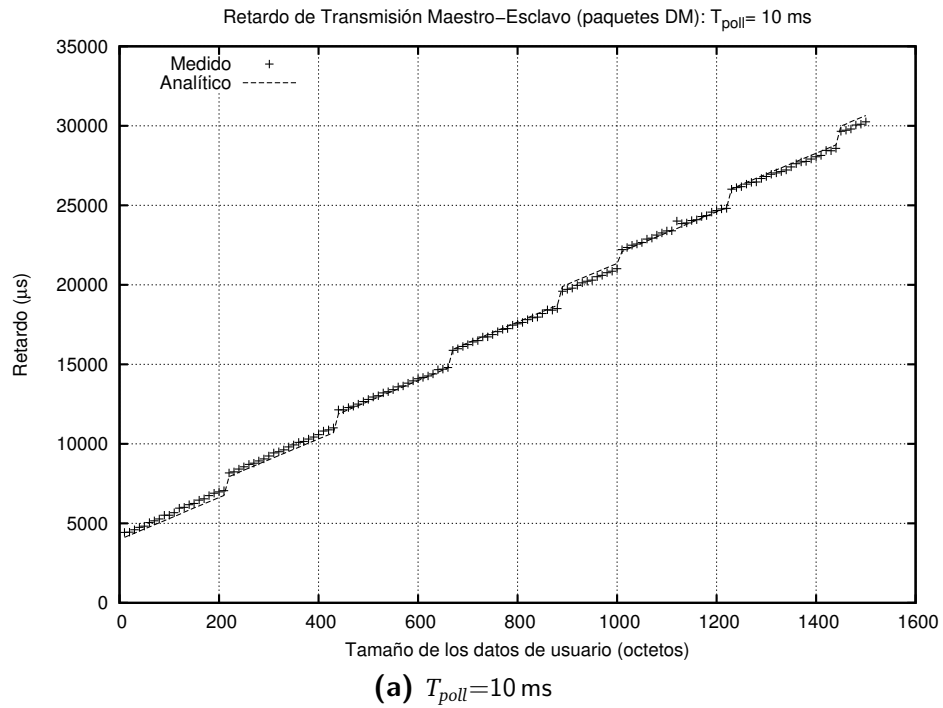


Figura 5.41.: Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DM (Revisión de HCI: 0x7a6).

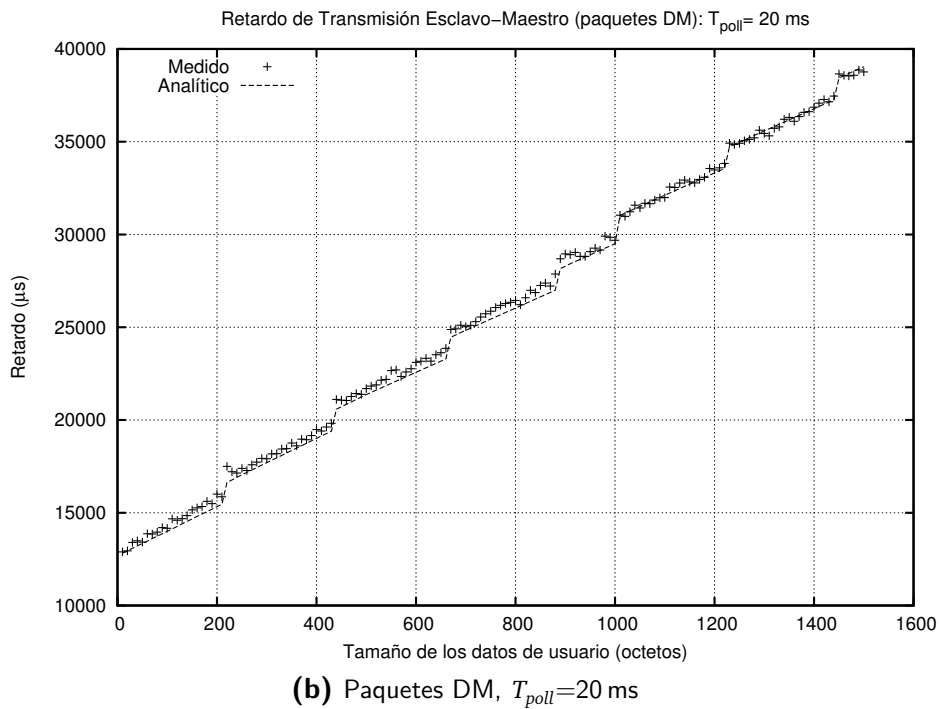
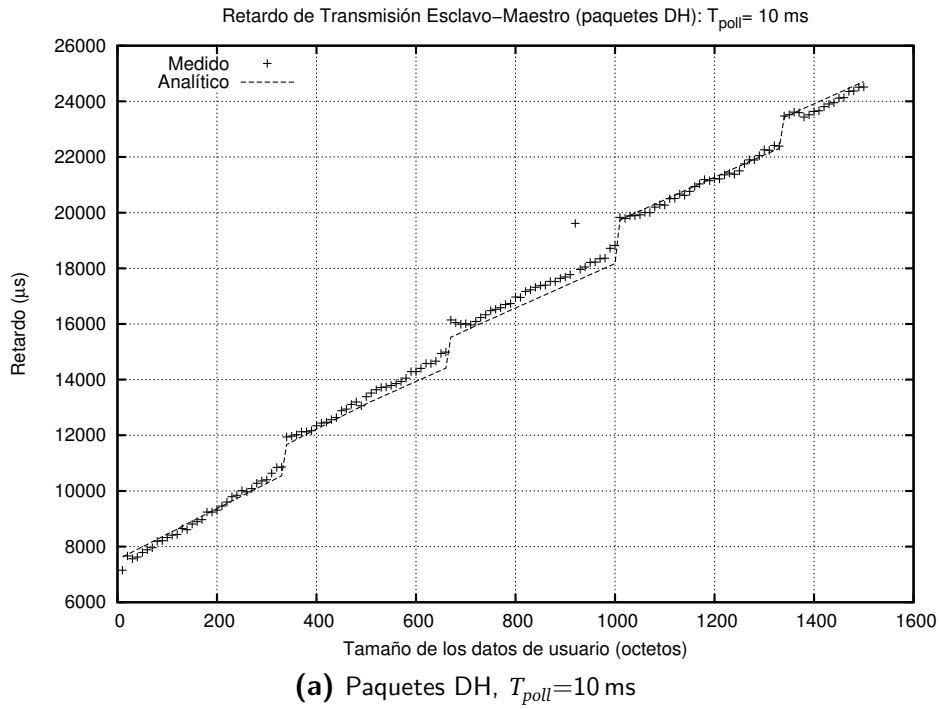


Figura 5.42.: Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido esclavo-maestro, para paquetes DH y DM del modo básico (Revisión de HCI: 0xc5c).

5.3.6.2. Paquetes DH en el modo EDR

En este apartado, para el modo EDR, se compara el retardo calculado a partir de la ecuación 5.18 con el medido, sin calidad de servicio, cuando el parámetro ACL MTU toma los valores de 1021, 740, 680, 512, 420, 384, 350, 310 y 256 octetos en el rango de longitudes comprendido entre 10 y 1500 octetos. Los resultados se muestran en las gráficas de las figuras 5.43, 5.44a, 5.44b, 5.45a, 5.45b, 5.46a, 5.46b, 5.47a y 5.47b.

Puede observarse que, en los casos en lo que el parámetro ACL MTU se configura a 740, 680 y 512 octetos, el retardo obtenido teóricamente se ajusta en el rango completo de longitudes al medido empíricamente. Sin embargo, tanto para el caso de 1021 octetos como para valores inferiores a 512 octetos puede apreciarse que en el intervalo de longitudes comprendidas entre los 1008 octetos y los 1008+367 octetos el retardo teórico subestima el medido, que presenta un comportamiento constante. Como hipótesis, cabe argumentar que es debido a un efecto de la capa de transporte que no ha sido factible caracterizar,

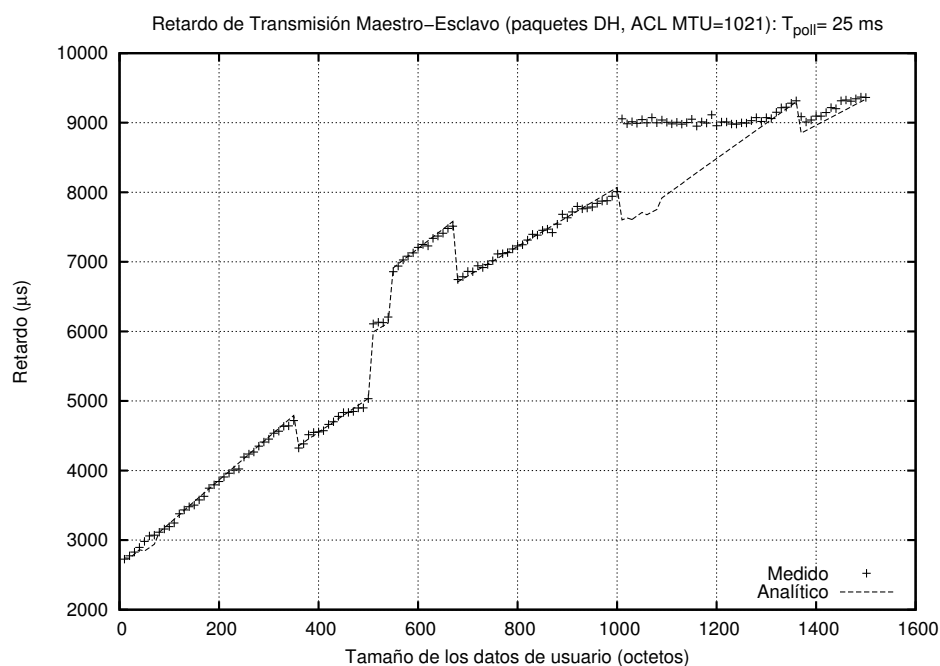


Figura 5.43.: Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DH y un valor del parámetro ACL MTU de 1021 octetos (Revisión de HCI: 0xc5c)

pero que tiene su origen en que el tiempo de transferencia por USB enmascara el retardo de transmisión Bluetooth. Para refrendar esta hipótesis, a modo de ejemplo, en las figuras 5.48a, 5.48b, 5.49a se muestra la mejora lograda, para los casos en los que el parámetro ACL MTU toma los valores 1021, 420 y 384, si se modifica la segunda rama de la ecuación 5.18 ($x > L_R$) según se indica a continuación:

$$t_{off} + n_{ff}(x) \cdot t_{ACK}^c(M(L_R)) + C_{ff}^{USB} + \quad (5.21)$$

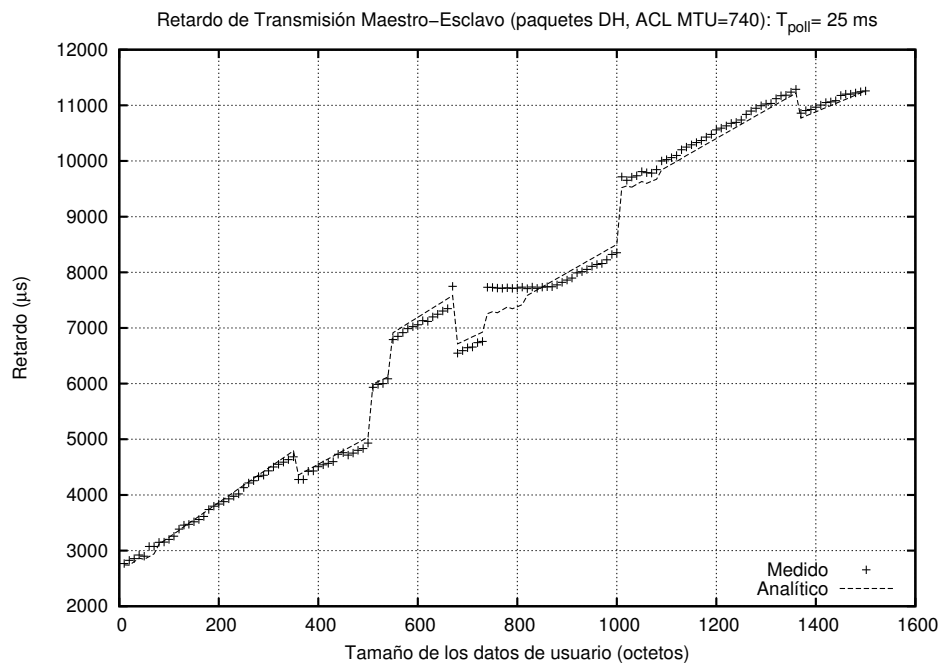
$$+ \max \left(t_{TX}(M(L_{ff}(x))) + C_{ff}^{USB}(M(L_{ff}(x))), t_{USB}(\min(M_{ACL}, M_{USB})) \right)$$

Sin embargo, esta corrección no da un resultado tan óptimo para el caso en el que el parámetro ACL MTU se configura a 310 octetos, según se ilustra en la figura 5.49b.

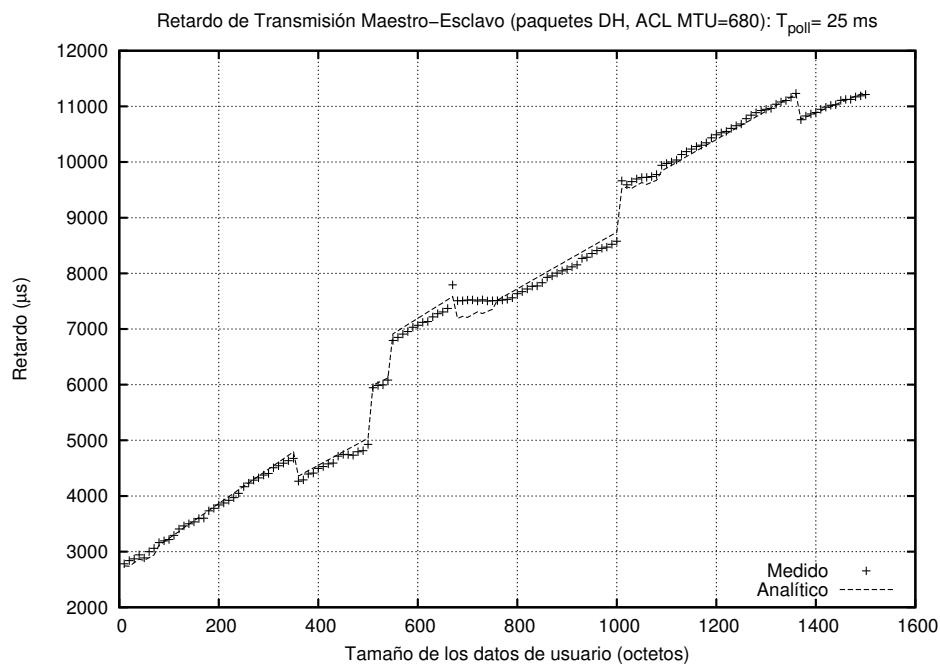
Nótese que la modificación introducida contempla el hecho de que el retardo de transmisión del segundo fragmento RFCOMM viene dado por el máximo entre dos valores:

- La suma del retardo de transmisión (t_{TX}) del segundo fragmento en cuestión y del retardo consumido en la transferencia USB de dicho fragmento (C_{ff}^{USB}).
- El retardo de transferencia USB del mínimo de M_{ACL} y M_{USB} .

No obstante, este último ajuste no puede ser justificado y la búsqueda de su sentido físico, relacionada con el modo en que tiene lugar la transferencia a través de la capa de transporte USB, escapa al ámbito de la presente tesis.

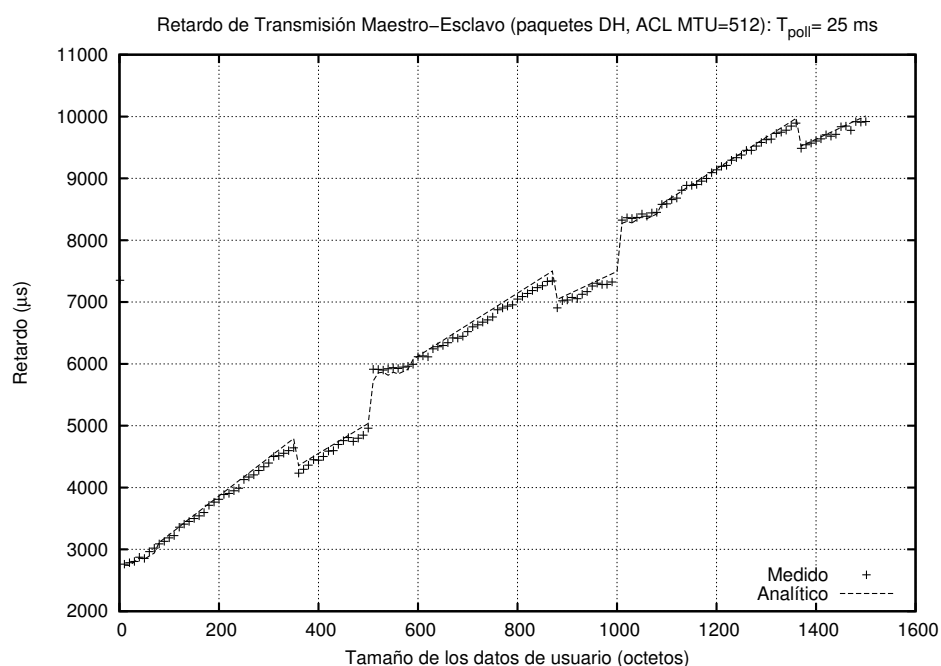


(a) ACL MTU de 740 octetos

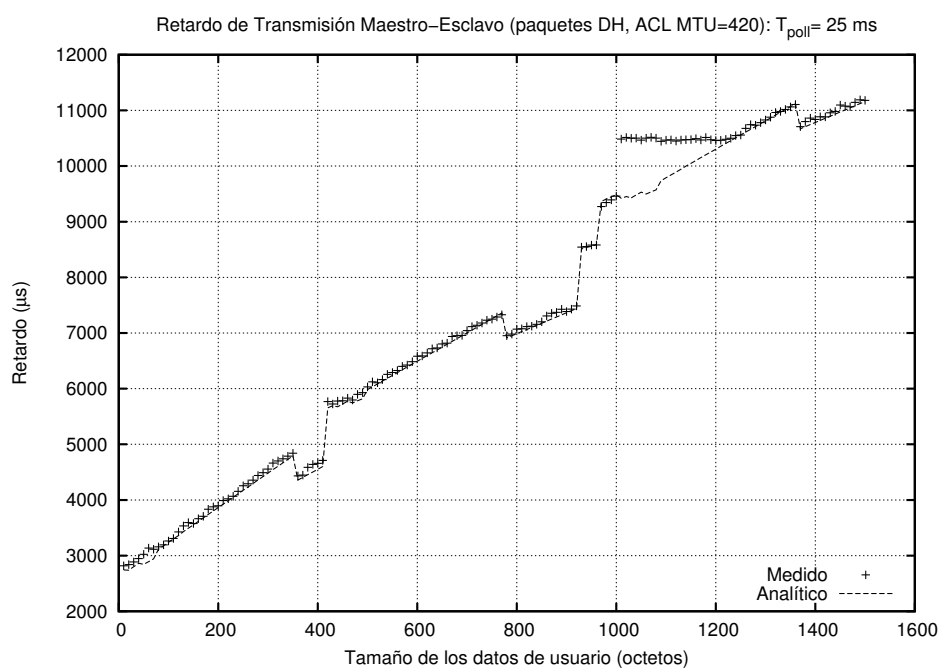


(b) ACL MTU de 680 octetos

Figura 5.44.: Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DH y valores del parámetro ACL MTU de 740 y 680 octetos (Revisión de HCI: 0xc5c)

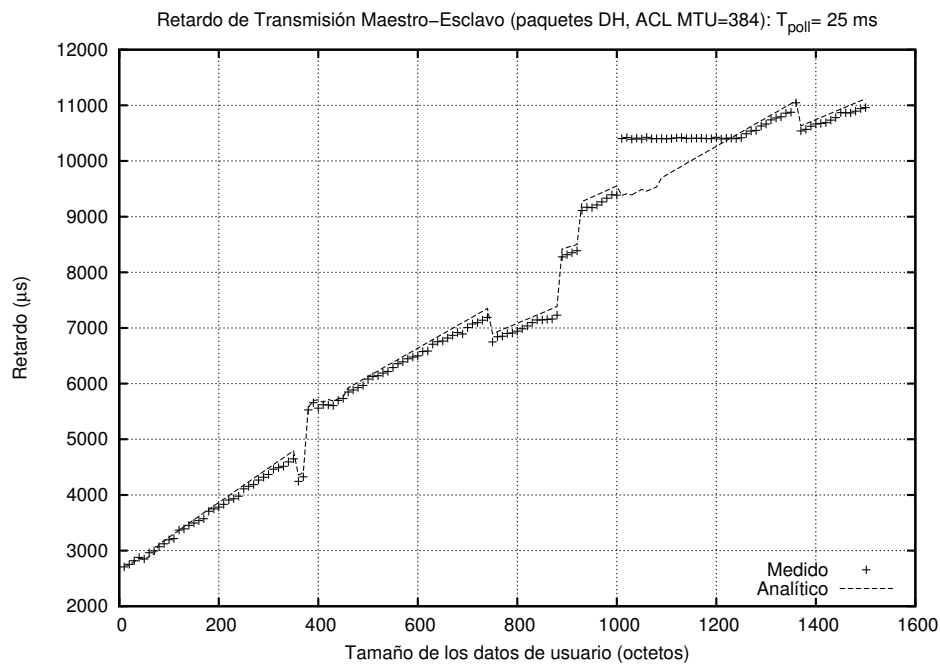


(a) ACL MTU de 512 octetos

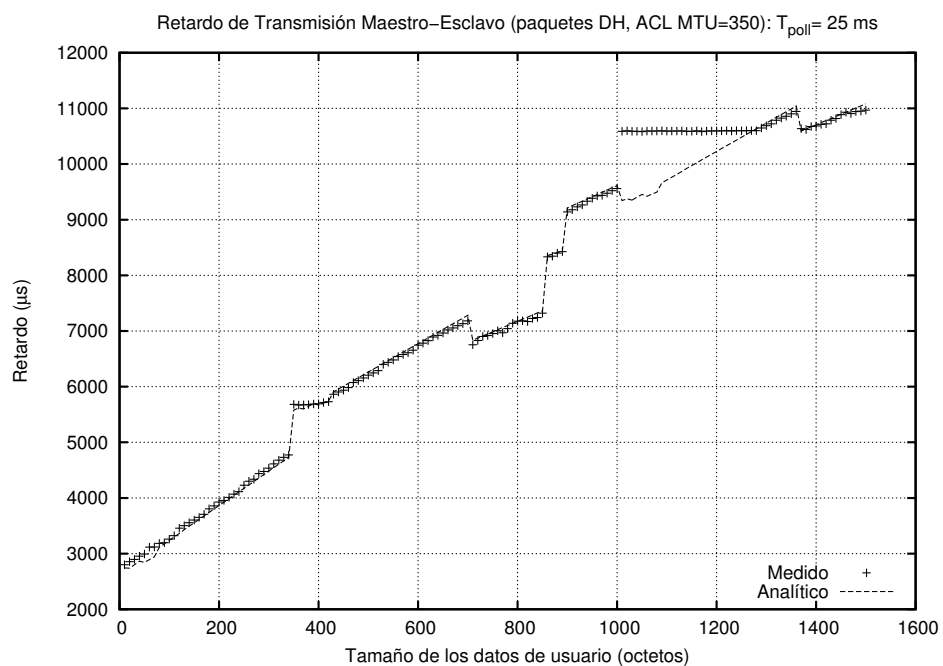


(b) ACL MTU de 420 octetos

Figura 5.45.: Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DH y valores del parámetro ACL MTU de 512 y 420 octetos (Revisión de HCI: 0xc5c)

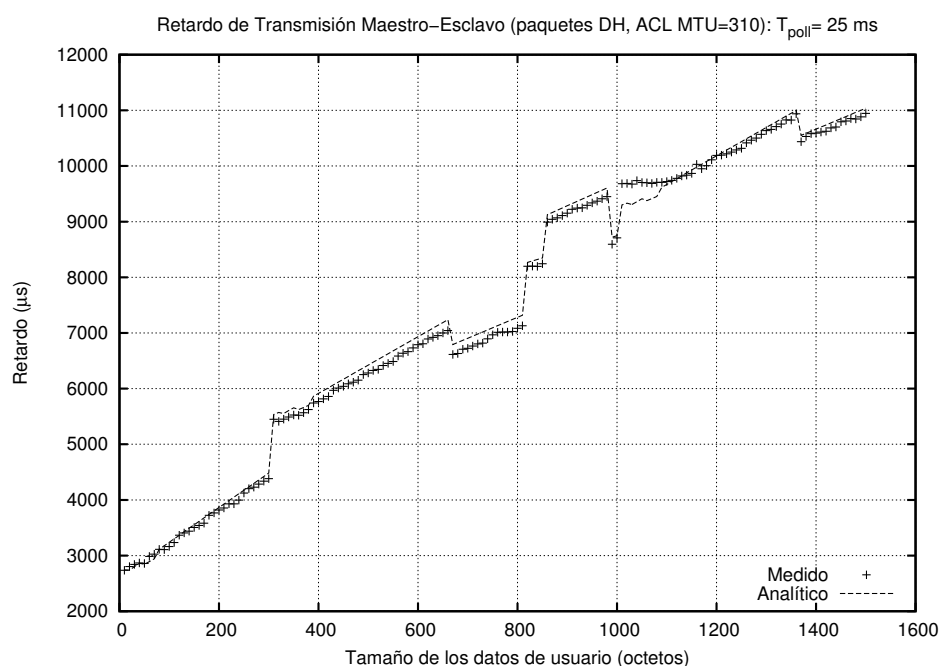


(a) ACL MTU de 384 octetos

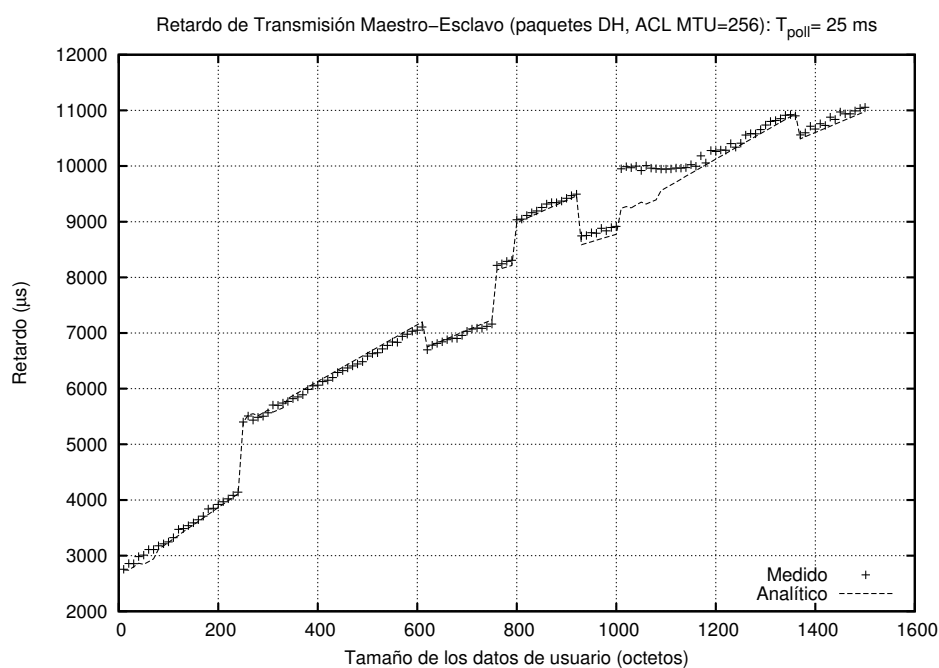


(b) ACL MTU de 350 octetos

Figura 5.46.: Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DH y valores del parámetro ACL MTU de 384 y 350 octetos (Revisión de HCI: 0xc5c)

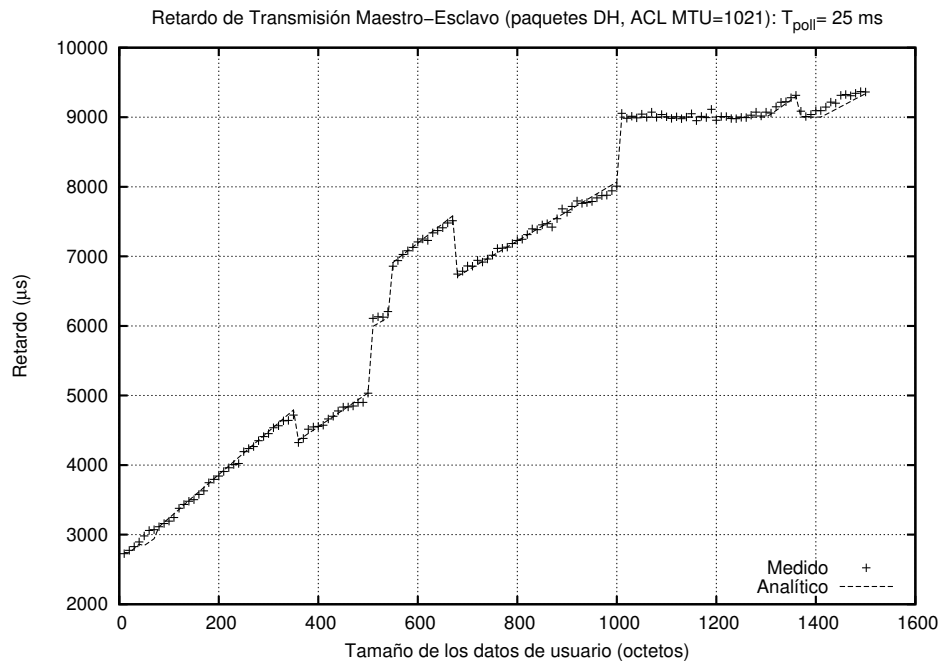


(a) ACL MTU de 310 octetos

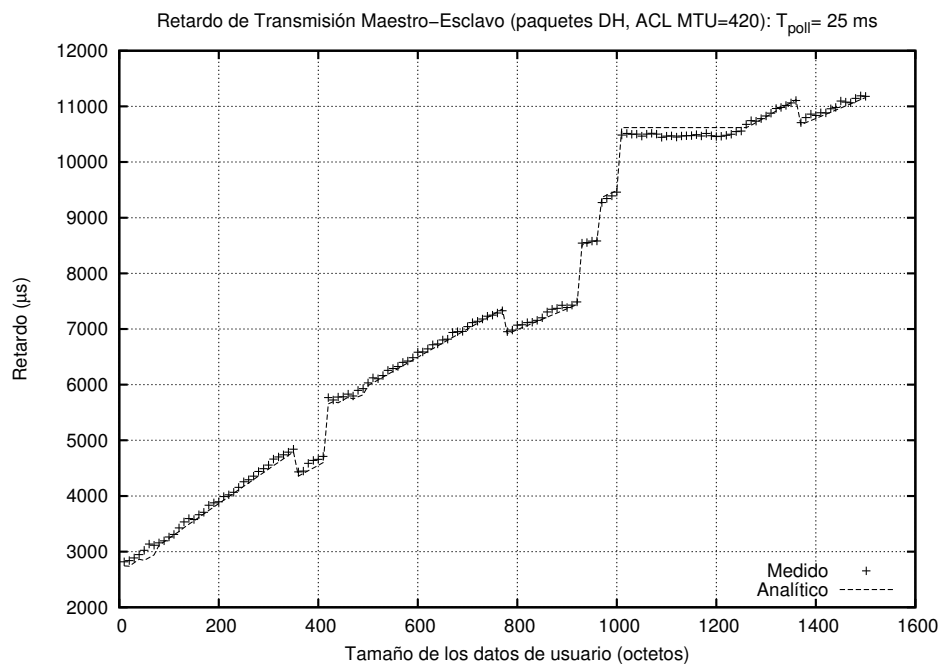


(b) ACL MTU de 256 octetos

Figura 5.47.: Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18, en el sentido maestro-esclavo, para paquetes DH y valores del parámetro ACL MTU de 310 y 256 octetos (Revisión de HCI: 0xc5c)

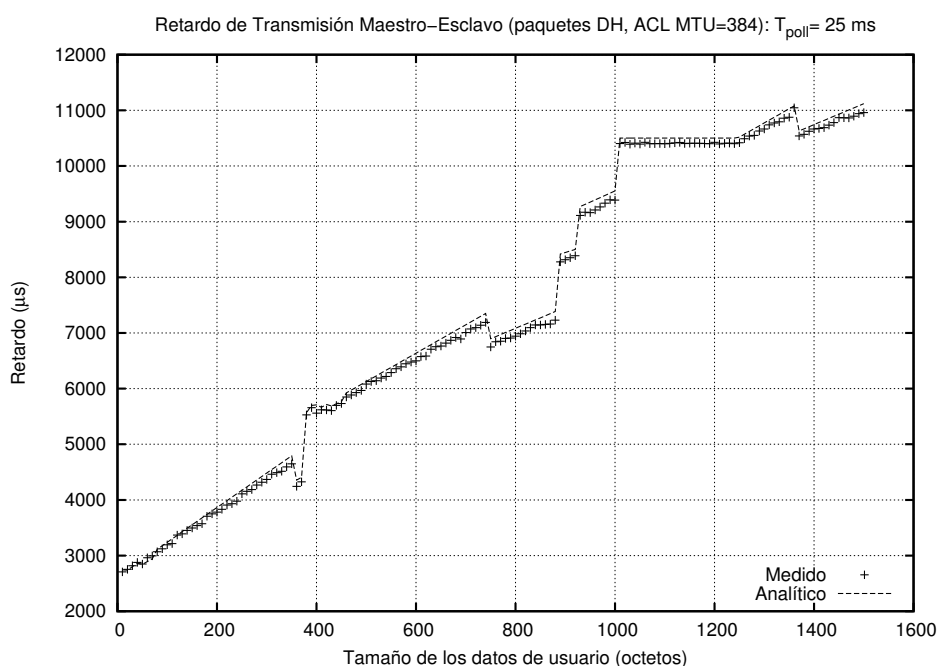


(a) ACL MTU de 1021 octetos

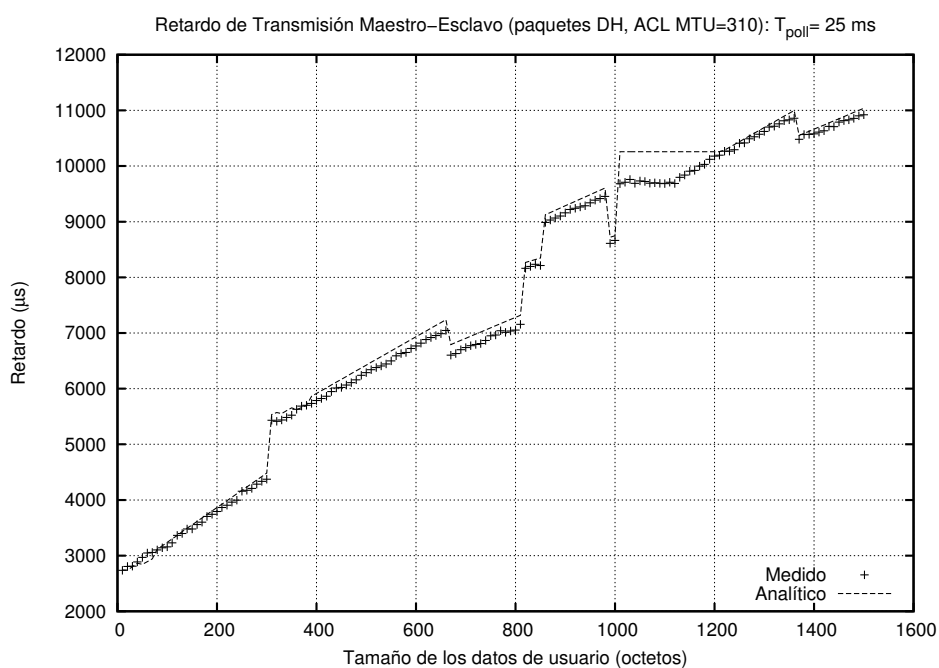


(b) ACL MTU de 420 octetos

Figura 5.48.: Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18 con la corrección dada por 5.18, en el sentido maestro-esclavo, para paquetes DH y valores del parámetro ACL MTU de 1021 y 420 octetos (Revisión de HCI: 0xc5c)



(a) ACL MTU de 384 octetos



(b) ACL MTU de 310 octetos

Figura 5.49.: Comparación del retardo medido y el teórico calculado a partir de la ecuación 5.18 con la corrección dada por 5.18, en el sentido maestro-esclavo, para paquetes DH y valores del parámetro ACL MTU de 384 y 310 octetos (Revisión de HCI: 0xc5c)

5.3.7. Recapitulación y conclusiones

En este apartado se recopilan las conclusiones extraídas sobre el comportamiento de los dispositivos CSR y, al mismo tiempo, se sintetizan las correcciones efectuadas para lograr modelar todos los fenómenos detectados que, en una primera aproximación, parecían enmascarar la validez del modelo desarrollado en el capítulo 4. Concretamente, el modelo ha tenido que ser extendido con el fin de tener en cuenta las siguientes consideraciones:

- En el sentido maestro-esclavo no existe dependencia del periodo T_{poll} configurado.
- La influencia sobre el retardo de transmisión de dos efectos:
 - El retardo asociado a la capa de transporte USB, independiente del sentido de transmisión, presente tanto en dispositivos Bluetooth 2.0+ EDR con versión de *firmware* 0x7a6 y 0xc5c, como en dispositivos Bluetooth 2.1 + EDR.
 - El retardo introducido por la fragmentación realizada de acuerdo con el valor del parámetro ACL MTU, efecto que tiene lugar tanto en el modo básico como en el modo EDR, para el sentido de transmisión maestro-esclavo y en dispositivos Bluetooth 2.0 + EDR a partir de la versión de *firmware* 0xc5c.

El proceso que se ha seguido para la modificación del modelo consta de los siguientes pasos:

1. Caracterización de la capa de transporte HCTL USB.
2. Cálculo de las componentes básicas del retardo de transmisión teniendo en cuenta el efecto de la capa de transporte HCTL USB.
3. Cálculo de las componentes básicas del retardo de transmisión considerando el efecto del parámetro ACL MTU.
4. Cálculo de las componentes básicas del retardo de transmisión teniendo en cuenta el efecto del parámetro ACL MTU y de la transferencia USB.
5. Cálculo del retardo a nivel de RFCOMM.

En los siguientes epígrafes se resumen cada una de estas etapas y, por último, se finaliza con las conclusiones inferidas a lo largo de todo el proceso.

Caracterización de la capa de transporte HCTL USB

Para evaluar el efecto sobre el retardo de la transferencia por USB ha sido necesario caracterizar la capa de transporte, según se describe en el apartado 5.3.3, concluyendo que el retardo introducido por dicha capa se puede aproximar, empíricamente, mediante la siguiente ecuación:

$$t_{USB}(x) = \begin{cases} 1,16 \cdot 10^3 + 2,19 \cdot x & x \leq M_{USB} \\ 2,39 \cdot 10^3 + 1,57 \cdot x & x > M_{USB} \end{cases} \quad (5.22)$$

donde se recuerda que M_{USB} , 512 octetos, denota el límite en el que tiene lugar un incremento del retardo de 1,20 ms y, que por tanto, delimita dos tramos con pendientes distintas.

Cálculo de las componentes básicas del retardo de transmisión considerando el efecto de la capa de transporte HCTL USB

En una primera aproximación, la inclusión en el modelo teórico del efecto de la capa de transporte se ha llevado a cabo definiendo las componentes $t_{TX}^c(x)$ y $t_{ACK}^c(x)$ a partir de las componentes básicas del retardo, t_{ACK} y $t_{TX}(x)$, tal y como se indica a continuación:

$$t_{TX}^c(x) = t_{TX}(x) + C_{TX}^{USB}(x) \quad (5.23)$$

$$t_{ACK}^c(x) = t_{ACK}(x) + C_{TX}^{USB}(x) \quad (5.24)$$

donde el factor de corrección $C_{TX}^{USB}(x)$ representa el efecto de la capa de transporte, y se define de forma diferente para cada uno de los modos de operación:

- Modo EDR:

$$C_{TX}^{USB}(x) = t_{USB}(x) \quad (5.25)$$

- Modo básico:

$$C_{TX}^{USB}(x) = \begin{cases} \frac{1}{2}t_{USB}(x) & 0 \leq x \leq M_{USB} \\ \frac{1}{2}t_{USB}(0) + \frac{1}{2}t_{USB}(x - M_{USB}) & (x > M_{USB}) \end{cases} \quad (5.26)$$

Cálculo de las componentes básicas del retardo de transmisión teniendo en cuenta el efecto del parámetro ACL MTU

Con el fin de incluir el efecto de la fragmentación según el valor del parámetro ACL MTU sobre la transmisión a nivel BB se han definido las componentes $t_{ACK}^{ACL}(x)$ y $t_{TX}^{ACL}(x)$, basadas en las originales, t_{ACK} y $t_{TX}(x)$, según se indica a continuación:

$$t_{ACK}^{ACL}(x) = \begin{cases} t_{ACK}(x) & 0 \leq x \leq M_{BB} \\ t_{ACK}(M_{BB}) + t_{ACK}(x - M_{BB}) & \text{en otro caso} \end{cases} \quad (5.27)$$

$$t_{TX}^{ACL}(x) = \begin{cases} t_{TX}(x) & 0 \leq x \leq M_{BB} \\ t_{ACK}(M_{BB}) + t_{TX}(x - M_{BB}) & \text{en otro caso} \end{cases} \quad (5.28)$$

donde M_{BB} representa el mínimo entre el parámetro ACL MTU, M_{ACL} , y la máxima capacidad de un paquete del nivel BB de 5 slots, L_5 , que será 224 octetos y 339 octetos, para los paquetes DM5 y DH5 del modo básico, y 1021 octetos, para el paquete 3-DH5 del modo EDR:

$$M_{BB} = \min(M_{ACL}, L_5) \quad (5.29)$$

Nótese que las componentes $t_{ACK}^{ACL}(x)$ y $t_{TX}^{ACL}(x)$, calculan el retardo de transmisión de los datos del nivel L2CAP, computando el slot de confirmación ($t_{ACK}^{ACL}(x)$) y sin computarlo ($t_{TX}^{ACL}(x)$), de forma diferente dependiendo de la relación entre su longitud y el mínimo entre dos valores, el parámetro ACL MTU (M_{ACL}) y la máxima longitud de un paquete BB de 5 slots (L_5), M_{BB} , según se indica a continuación:

- Longitud inferior a M_{BB} octetos: El retardo previsto originalmente, t_{ACK} y $t_{TX}(x)$, obtenido mediante las ecuaciones 4.11 y 4.13.
- Longitud superior a M_{BB} octetos: Se estima que es la suma de dos componentes:
 1. El retardo de transmisión de M_{BB} octetos, considerando la confirmación del esclavo: $t_{ACK}(M_{BB})$, tanto si se computa $t_{ACK}^{ACL}(x)$ como $t_{TX}^{ACL}(x)$.
 2. El retardo de transmisión de los octetos restantes $-(x - M_{BB})$, calculado utilizando las componentes básicas, $t_{ACK}(x - M_{BB})$ y $t_{TX}(x - M_{BB})$, según se trate de $t_{ACK}^{ACL}(x)$ ó de $t_{TX}^{ACL}(x)$.

Cálculo de las componentes básicas del retardo de transmisión teniendo en cuenta el efecto del parámetro ACL MTU y de la transferencia USB

El primer paso para reflejar la combinación del efecto de la capa de transporte y de la fragmentación según el valor del parámetro ACL MTU se concreta en la redefinición de los términos $t_{ACK}^c(x)$ y $t_{TX}^c(x)$, tal y como se resume a continuación:

- $t_{ACK}^c(x)$: El retardo de transmisión, incluyendo el *slot* de confirmación del esclavo, no contempla el efecto de la transferencia USB, aunque sí tiene en cuenta la influencia del parámetro ACL MTU, coincidiendo por tanto su definición con la componente $t_{ACK}^{ACL}(x)$:

$$t_{ACK}^c(x) = t_{ACK}(x) \quad (5.30)$$

Es conveniente aclarar que el retardo asociado a la transferencia por la capa de transporte USB no puede integrarse en la componente $t_{ACK}^c(x)$ porque da lugar a una sobreestimación a partir de los 1008 octetos de datos de usuario.

- $t_{TX}^c(x)$: El retardo de transmisión sin computar el *slot* de confirmación se calcula a partir de la siguiente expresión:

$$t_{TX}^c(x) = t_{TX}^{ACL}(x) + C_{TX}^{USB}(x) \quad (5.31)$$

Así, la componente $t_{TX}^c(x)$ engloba el efecto de la fragmentación al tamaño del parámetro ACL MTU:

- En la transmisión Bluetooth, representado por la componente $t_{TX}^{ACL}(x)$.
- En la transferencia por la capa de transporte USB, representado por el término $C_{TX}^{USB}(x)$, que se define de forma diferente para cada uno de los modos de operación:
 - Modo básico:

$$C_{TX}^{USB}(x) = \begin{cases} \frac{1}{2}t_{USB}(x) & 0 \leq x \leq \min(M_{USB}, M_{ACL}) \\ \frac{1}{2}t_{USB}(M_{ACL}) & (x > M_{ACL}) \wedge (M_{ACL} \leq M_{USB}) \\ \frac{1}{2}t_{USB}(0) + \frac{1}{2}t_{USB}(x - M_{USB}) & (x > M_{USB}) \wedge (M_{ACL} > M_{USB}) \end{cases} \quad (5.32)$$

- Modo EDR:

$$C_{TX}^{USB}(x) = \begin{cases} t_{USB}(x) & 0 \leq x \leq M_{ACL} \\ \frac{1}{2}t_{USB}(\min(M_{ACL}, M_{USB})) + & x - M_{ACL} \leq M_{USB} \\ + \frac{1}{2}t_{USB}(x - \min(M_{ACL}, M_{USB})) & \\ \frac{1}{2}t_{USB}(M_{USB}) + t_{USB}(M_{ACL}) & \text{en otro caso} \end{cases} \quad (5.33)$$

Cálculo del retardo a nivel de RFCOMM

El segundo paso en la integración de la combinación de ambos efectos, el introducido por la capa de transporte y el debido a la fragmentación según el valor del parámetro ACL MTU, ha supuesto redefinir la ecuación que calcula el retardo de transmisión a nivel RFCOMM, $\overline{t}_R(x)$, tanto para el modo básico como para el modo EDR, a partir de la ecuación 5.18, que se reproduce a continuación:

$$\overline{t}_R(x) = \begin{cases} t_{off} + t_{TX}^c(M(x)) & x \leq L_R \\ t_{off} + n_{fuf}(x) \cdot t_{ACK}^c(M(L_R)) + C_{fuf}^{USB} + & x > L_R \\ + t_{TX}(M(L_{ff}(x))) + C_{ff}^{USB}(M(L_{ff}(x))) & \end{cases} \quad (5.34)$$

donde:

- El término t_{off} representa un *offset* promedio que incluye, en el sentido de transmisión esclavo-maestro, el *slot* en el que maestro direcciona al esclavo²:

$$t_{off} = \begin{cases} 3 \cdot T_S & \text{Sentido maestro-esclavo, paquetes DM ó DH} \\ 2 \cdot T_S & \text{Sentido maestro-esclavo, paquetes DH-EDR} \\ \frac{T_{poll}}{2} + 3 \cdot T_S & \text{Sentido esclavo-maestro, paquetes DM ó DH} \\ \frac{T_{poll}}{2} + T_S & \text{Sentido esclavo-maestro, paquetes DH-EDR} \end{cases} \quad (5.35)$$

- El factor de corrección, C_{ff}^{USB} , común a ambos modos de operación (básico y EDR), representa el retardo introducido por la capa de transporte para una trama RFCOMM inicial, considerando el efecto del parámetro ACL MTU, y se define a partir de la siguiente ecuación:

$$C_{ff}^{USB} = \begin{cases} \frac{t_{USB}(\min(M_{ACL}, M_{USB}))}{2} & M_{ACL} \leq L_5 \\ \frac{t_{USB}(\min(M_{USB}, L_5)) + t_{USB}(0)}{2} & M_{ACL} > L_5 \end{cases} \quad (5.36)$$

- El término $C_{ff}^{USB}(x)$ representa el retardo introducido por la capa de transporte para una trama RFCOMM final, cuya definición difiere según el modo de operación:

- Modo básico: El término $C_{ff}^{USB}(x)$ es nulo: $C_{ff}^{USB}(x) = 0$
- Modo EDR: $C_{ff}^{USB}(x) = \frac{t_{USB}(x)}{2}$

Obsérvese que la ecuación que calcula el retardo a nivel de RFCOMM lo hace distinguiendo que la longitud de los datos de usuario sea inferior o superior a la máxima capacidad de una trama RFCOMM ($L_R = 1008$ octetos), de forma que si los datos de usuario:

- No superan la longitud de una trama RFCOMM, $L_R = 1008$ octetos: El retardo de transmisión viene dado por el término $t_{TX}^c(x)$, que integra la combinación del efecto de la capa de transporte y de la fragmentación según el parámetro ACL MTU.

²Válido para la versión de *firmware* 0xc5c, versión comercial más extendida

- Exceden la longitud de una trama RFCOMM, $L_R = 1008$ octetos: Se ha podido concluir que:
 - Para el primer fragmento RFCOMM inicial (no único) hay que contemplar la suma de dos contribuciones:
 - El retardo de transmisión, incluyendo el *slot* de confirmación del esclavo, que se obtiene a partir de la ecuación $t_{ACK}^c(x)$, incluyendo el efecto de la fragmentación según el parámetro ACL MTU.
 - El retardo asociado a la transferencia por la capa de transporte USB, que se computa mediante el factor de corrección: C_{ff}^{USB} .
 - Para el fragmento RFCOMM de continuación: Al retardo de transmisión Bluetooth, $t_{TX}(x)$, se suma el factor de corrección, $C_{ff}^{USB}(x)$, que representa el retardo introducido por la capa de transporte para una trama RFCOMM final, siendo nulo en el modo básico.

Conclusiones

Finalmente, cabe resaltar las siguientes conclusiones extraídas a partir del análisis realizado:

- La fragmentación en bloques de 512 octetos, realizada para la transferencia USB, no influye en el esquema de transmisión, ya que afecta únicamente a la comunicación entre el *Host* y el *Host Controller*.
- La fragmentación que se lleva a cabo atendiendo al valor del parámetro ACL MTU (M_{ACL}) condiciona:
 - La transferencia a través de la capa de transporte.

- Los paquetes BB utilizados en la transmisión Bluetooth, de forma que si la longitud de los datos a transmitir es superior al valor del parámetro ACL MTU y, esta a su vez, es inferior a la máxima capacidad de un paquete de 5 *slots*, no se va a conseguir el máximo rendimiento porque se va a transmitir, un primer paquete con M_{ACL} octetos (ni siquiera completo) y un segundo paquete (o incluso más) con el número de octetos en los que la longitud de los datos de usuario excede el límite impuesto por el parámetro ACL MTU.
- La principal consecuencia de la combinación de los efectos —fragmentación al valor del parámetro ACL MTU y transferencia por USB— es una pérdida de rendimiento. Concretamente, para maximizar el rendimiento, minimizando el retardo de transmisión, para cada uno de los modos se han de cumplir las siguientes condiciones:
 - Modo básico: El parámetro ACL MTU (M_{ACL}) se configura a un valor igual o superior a la máxima capacidad de un paquete de 5 *slots* (L_5): $M_{ACL} \geq L_5$.
 - Modo EDR: Los parámetros M_{ACL} y M_{USB} coinciden, es decir, si $M_{ACL} = M_{USB} = 512$ octetos.

5.4. Dispositivos ISS

Se han repetido gran parte de las medidas anteriores utilizando dispositivos del fabricante ISS [ISSC] con las siguientes versiones y revisiones del *firmware* HCI y del nivel LM:

- *Versión de HCI: 2.0 (0x3); Revisión de HCI: 0x2da*
- *Versión del nivel LM: 2.0 (0x3); Subversión del nivel LM: 0x2da*

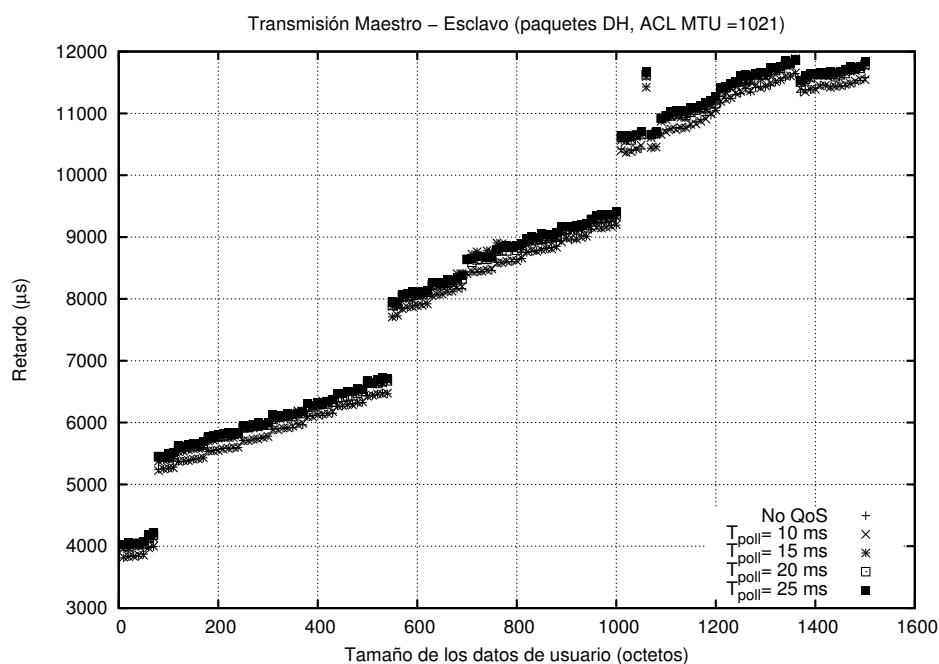
Al igual que con los dispositivos CSR, para evaluar el efecto de la calidad de servicio, una vez establecida la conexión RFCOMM, también se ha utilizado el comando *HCI_QoS_Setup* especificando *Best Effort* como tipo de servicio y los valores indicados en la tabla 5.3 para el parámetro *Latency*. El retardo medio medido en cada uno de los sentidos de transmisión, configurando el uso de paquetes DH del modo básico y EDR y para un valor del parámetro ACL MTU³ de 1021 octetos, se presenta en las figuras 5.50a y 5.50b. Puede observarse que:

- En el sentido maestro-esclavo: El resultado es el mismo en todos los casos, independientemente de que se configure o no la calidad de servicio, y del valor del parámetro *Latency* especificado.
- En el sentido esclavo-maestro: Se alcanzan retardos de transmisión muy elevados. De hecho, incluso cuando el parámetro *Latency* se configura a un valor de 5 ms, el tiempo necesario para la transmisión de 10 octetos es de 15 ms, valor que aumenta hasta los 45 ms cuando el parámetro *Latency* se fija a un valor de 25 ms, frente a los 4 ms consumidos en el sentido maestro-esclavo.

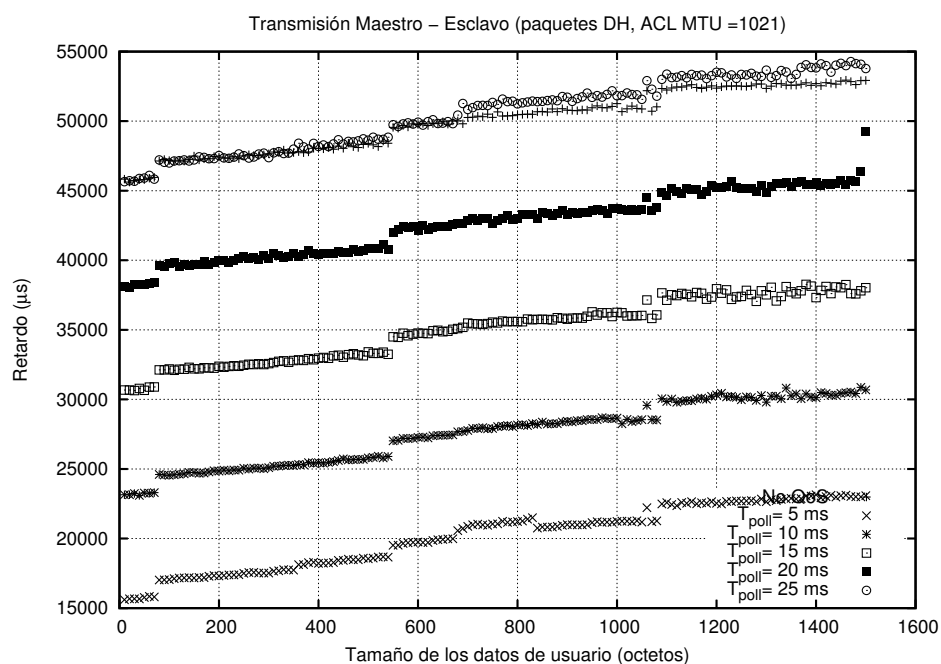
A tenor de los resultados, el resto del apartado se centra en analizar el comportamiento en el sentido de transmisión maestro-esclavo, ya que para este caso el retardo alcanzado se encuentra dentro de unos límites razonables.

De la comparación del retardo medido en el sentido de transmisión maestro-esclavo con el previsto teóricamente mediante la ecuación 4.1, llama la atención que, a diferencia

³En los dispositivos Bluetooth del fabricante ISS no es posible modificar el valor del parámetro ACL MTU como en el caso de los dispositivos CSR, dado que la herramienta *bccmd*, integrada en BlueZ y utilizada para modificarlo, sólo implementa comandos específicos del fabricante CSR.



(a) Sentido de transmisión maestro-esclavo



(b) Sentido de transmisión esclavo-maestro

Figura 5.50.: Efecto de la configuración de la calidad de servicio en dispositivos ISS, para paquetes DH y un valor del parámetro ACL MTU de 1021 octetos.

de lo que sucede con dispositivos del fabricante CSR, no se reflejan los cambios de velocidad de modulación en la curva del retardo.

Concretamente, en la figura 5.51a, puede apreciarse que una vez que se excede la capacidad de un paquete 2-DH3 (367 octetos), no se produce un salto descendente como resultado del cambio de la modulación $\pi/4$ DQPSK a 8 DPSK. Análogamente, tampoco tiene lugar decremento alguno en el caso en el que, teóricamente, se aumenta la tasa de 2 Mbps a 3 Mbps, cuando se pasa de emplear un paquete 2-DH5 a un paquete 3-DH5. Por contra, cabe destacar que los saltos ascendentes se producen en torno a las longitudes a las que se llenan los paquetes de máxima velocidad para un determinado número de *slots*: 83 octetos (3-DH1), 552 octetos (3-DH3) y 1021 octetos (3-DH5).

Al respecto, es conveniente aclarar que la razón que justifica estos resultados no es la misma en los dos primeros casos que en el último, ya que el agotamiento de la capacidad de un paquete 3-DH5 coincide con el tamaño de los datos de usuario cuyo envío pasa a requerir dos tramas RFCOMM, punto a partir del cual, el comportamiento real sólo se diferencia del previsto teóricamente en cierto *offset*.

Sin embargo, en los dos primeros casos, cabe suponer que el módulo radio no entrega los datos al *Host* Bluetooth hasta que no se completa el número de *slots* ocupados por el paquete en el que se reciben los datos. Con el fin de incorporar en el modelo este comportamiento se ha redefinido la ecuación que calcula el retardo a nivel de RFCOMM:

$$\overline{t}_R(x) = \begin{cases} t_{off} + t_{TX_{ISS}}^{DH}(M(x)) & x \leq L_R \\ t_{off} + n_{nff}(x) \cdot t_{ACK}(M(L_R)) + t_{TX}(M(L_{ff}(x))) & x > L_R \end{cases} \quad (5.37)$$

siendo:

$$t_{TX_{ISS}}^{DH}(x) = \begin{cases} 0 & x = 0 \\ T_S & 0 < x \leq L_1 \\ 3 \cdot T_S & L_1 < x \leq L_3 \\ 5 \cdot T_S & L_3 < x \leq L_5 \\ t_{ACK} \cdot (L_5) \cdot \left\lfloor \frac{x}{L_5} \right\rfloor + & x > L_5 \\ + t_{TX_{ISS}}^{DH} \cdot (x \bmod L_5) & \end{cases} \quad (5.38)$$

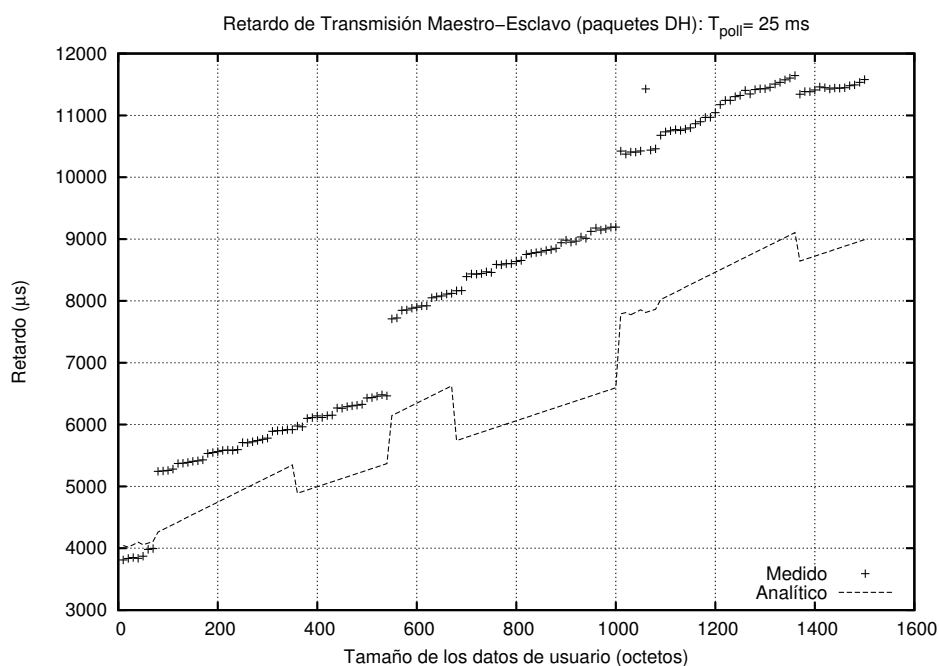
donde:

- L_1 , L_3 y L_5 , al igual que en la ecuación 4.11, denotan las longitudes máximas del *payload* de un paquete del nivel BB de 1, 3 ó 5 *slots* que, para los paquetes DH EDR de máxima velocidad (3 Mbps) —3-DH1, 3-DH3 y 3-DH5— son, respectivamente, 83, 552 y 1021 octetos.
- El término designado como t_{off} representa un *offset* promedio de $4 \cdot T_s$ (correspondiente a un intervalo de *poll* de 5 ms) que, tal y como las medidas ponen de manifiesto, resulta ser independiente del valor del parámetro *Latency* configurado.

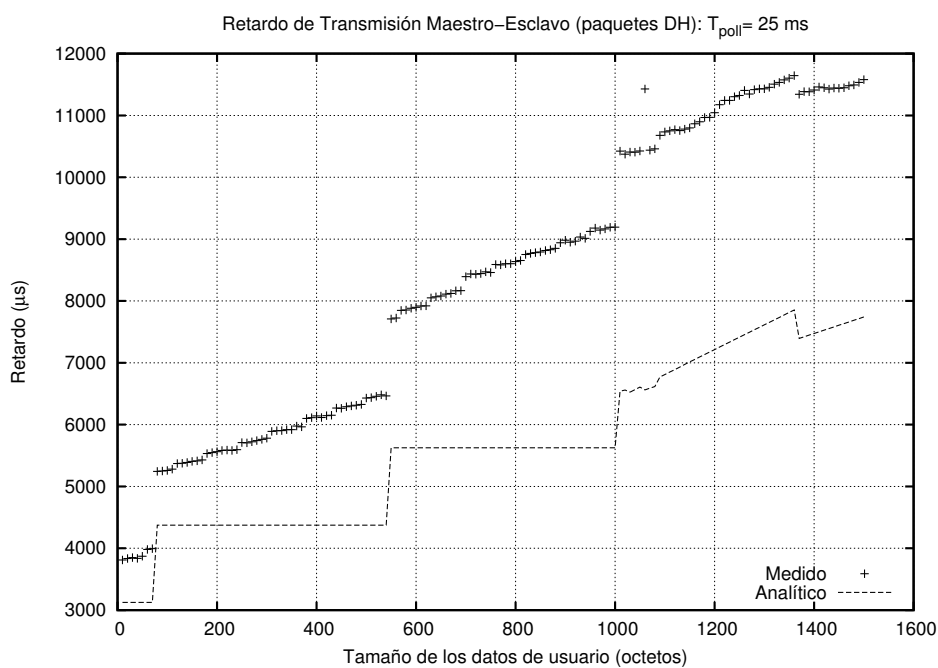
Nótese que, en la ecuación 5.37, la primera rama determina el tiempo necesario para la transmisión de datos de usuario en una única trama RFCOMM, considerando que el cómputo se realiza en términos de *slots* completos, mientras que la segunda asume que los datos transportados en una segunda trama responden al comportamiento original, dado por la componente $t_{TX}(x)$, que calcula el retardo en función del número de bits transmitidos.

Contra la hipótesis planteada existe un argumento que, en principio, podría considerarse que la invalida: los tramos entre saltos deberían presentar un retardo constante, según se ilustra en la figura 5.51b, donde la curva teórica ha sido calculada a partir de la ecuación 5.37. Sin embargo, el comportamiento creciente observado cuando cabe esperar un retardo constante puede justificarse teniendo en cuenta, tal y como se ha demostrado con los dispositivos CSR, que la medida realizada incluye un efecto de la capa de transporte. No obstante, en este caso, no ha sido factible caracterizar el retardo introducido por la capa HCTL, debido a que los dispositivos ISS al enviar el comando *HCI_Write_Loopback_Mode* retornan un error que, según se indica en el estándar Bluetooth [Bluetooth SIG04, Bluetooth SIG07, vol. 2, parte D], significa que el comando solicitado puede no estar implementado.

Por consiguiente, además de la ausencia de indicios de cambios de modulación explicada en el epígrafe anterior, la única evidencia que se puede aportar para ratificar que, como efecto de la capa de transporte, el retardo medido no responde a la curva obtenida a partir de la ecuación 5.37, es el resultado de restar al retardo medido el retardo calculado mediante la ecuación 5.37, que se muestra en la figura 5.52. En la citada figura puede apreciarse que para longitudes inferiores a 1008 octetos la diferencia tiene un comportamiento lineal,



(a) Retardo teórico calculado mediante la ecuación 4.1



(b) Retardo teórico previsto mediante la ecuación 5.37

Figura 5.51.: Comparación del retardo medio de transmisión teórico a partir de la ecuaciones 4.1 y 5.37 con el medido para dispositivos ISS en el sentido maestro-esclavo.

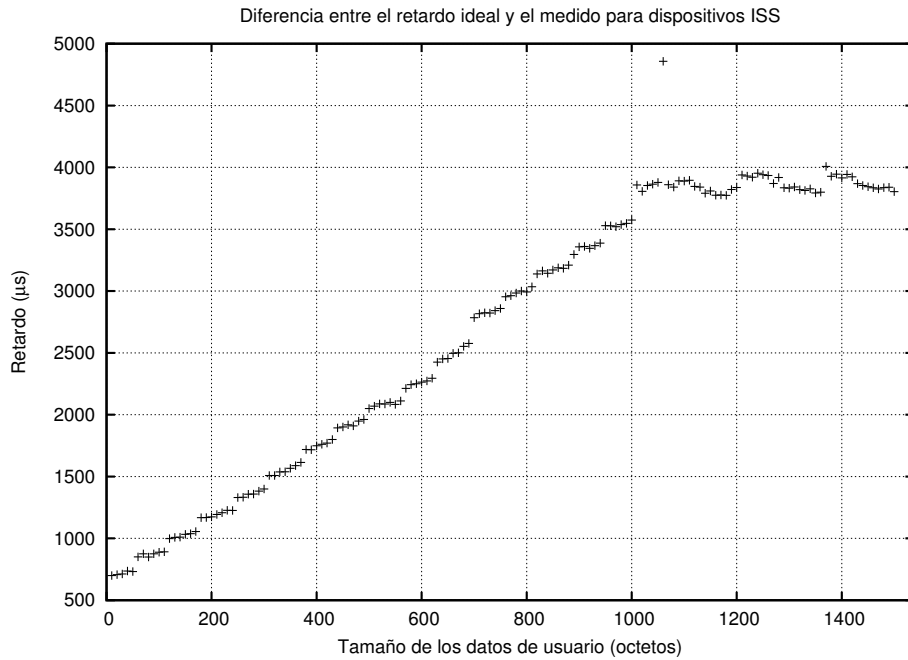


Figura 5.52.: Diferencia entre el retardo medido y el retardo calculado mediante la ecuación 5.37

mientras que para longitudes superiores, la diferencia es un valor constante. Así, a partir del retardo medio, empleando el método de mínimos cuadrados, se ha obtenido la recta de regresión correspondiente al tramo lineal, resultando la siguiente fórmula empírica para el rango completo de longitudes:

$$t_{USB}(x) = \begin{cases} 557 + 3,02 \cdot x & x \leq L_R \\ 3,60 \cdot 10^3 & x > L_R \end{cases} \quad (5.39)$$

Nótese que, en este caso, a diferencia de lo que sucede con los dispositivos del fabricante CSR, el retardo introducido por la capa de transporte no presenta discontinuidad alguna en la longitud de 512 octetos (véase la figura 5.21b).

En la figura 5.53 se compara el retardo medido con el calculado a partir de la ecuación 5.37 a la que se le suma el efecto de la capa de transporte dado por la ecuación 5.39.

Finalmente, en lo que respecta a las medidas que se obtienen cuando se configuran los dispositivos ISS para utilizar exclusivamente paquetes DM, tan solo cabe concluir que

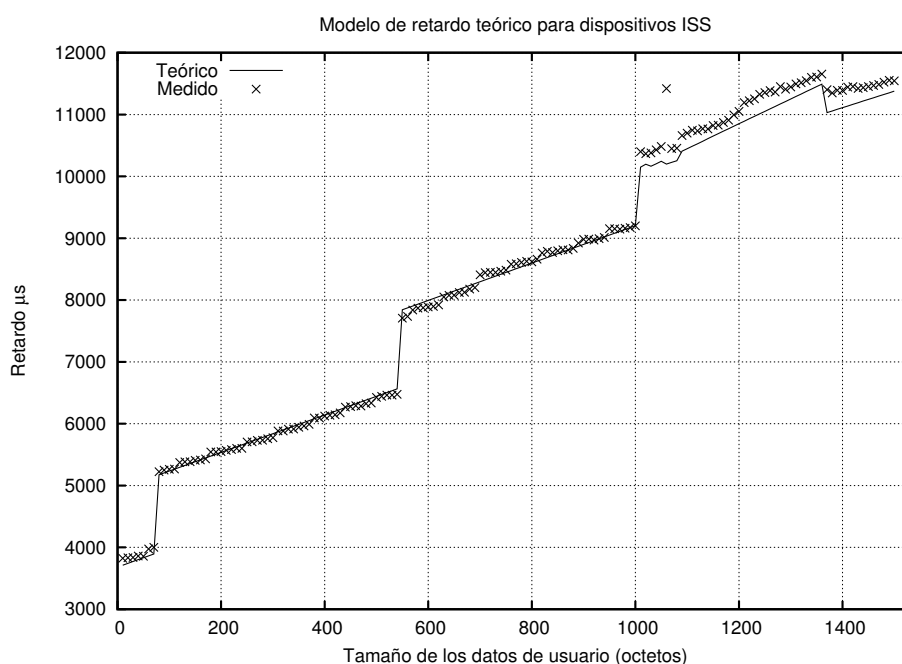
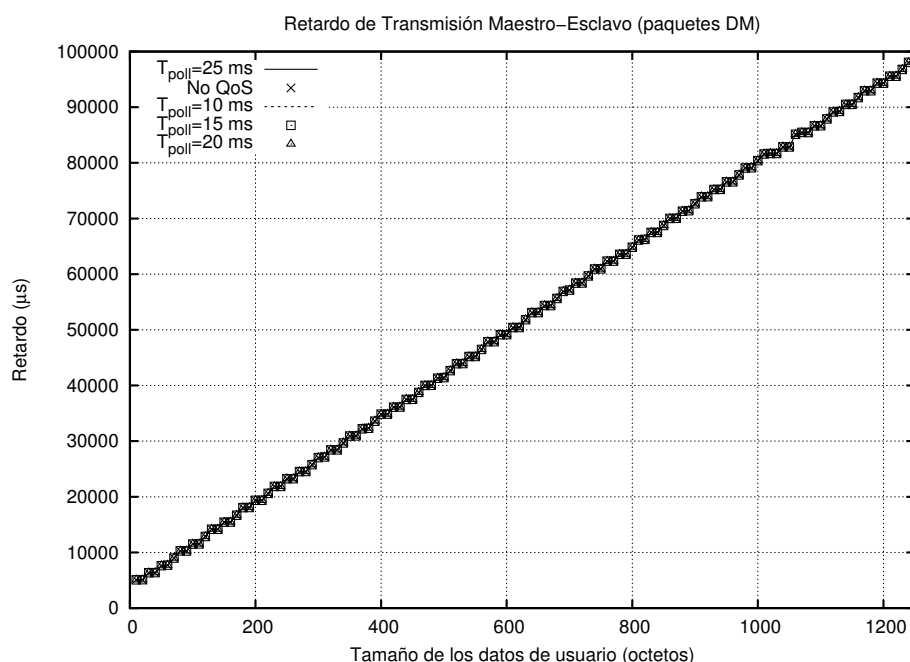
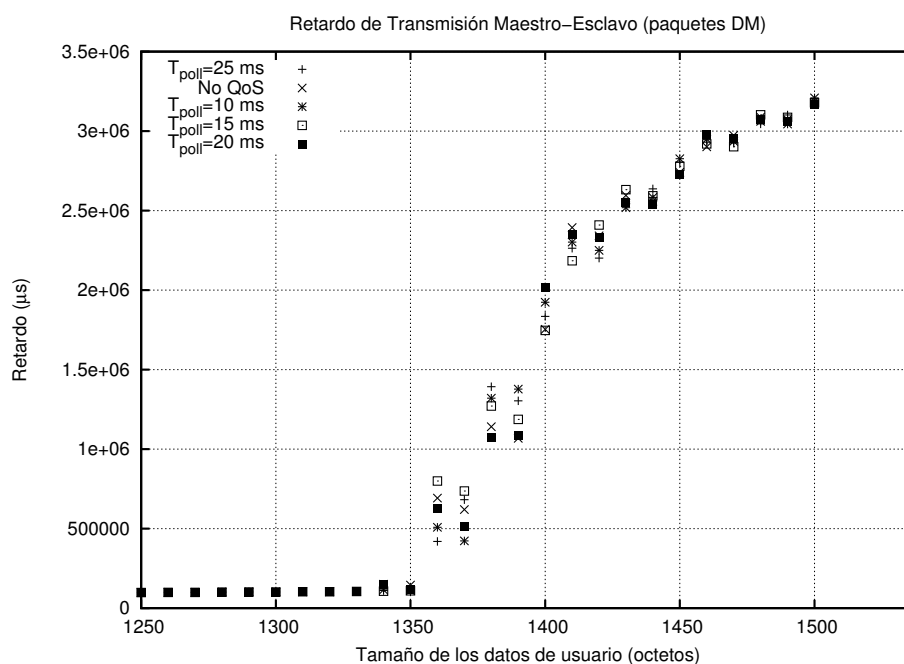


Figura 5.53.: Comparación entre el retardo medido y el retardo teórico calculado a partir de la ecuación 5.37, teniendo en cuenta la estimación del retardo introducido por la capa de transporte HCTL.

no son operativos debido a la anormalidad manifiesta del comportamiento observado, tal y como se muestra en la figura 5.54a para el rango de longitudes comprendido entre 10 y 1250 octetos, y en la figura 5.54b para el intervalo restante 1250-1500 octetos, en el que el retardo degenera de forma significativa, debido al efecto de encolamiento que se produce en cuanto el retardo de transmisión alcanza el intervalo entre envíos consecutivos: 100 ms. Es decir, con paquetes DM y en un escenario sin pérdidas, los dispositivos ISS presentan retardos superiores a 100 ms para el envío de más de 1250 octetos de usuario.



(a) Rango de longitudes comprendido entre 10 y 1250 octetos



(b) Rango de longitudes comprendido entre 1250 y 1500 octetos

Figura 5.54.: Retardos de transmisión obtenidos con dispositivos del fabricante ISS en el sentido de transmisión maestro-esclavo, empleando paquetes DM.

5.5. Dispositivos *Broadcom*

Se han repetido las medidas utilizando dispositivos del fabricante *Broadcom Corporation* con las siguientes versiones y revisiones del *firmware* HCI y del nivel LM:

- Versión de HCI: 2.0 (0x3); Revisión de HCI: 0x4000
- Versión del nivel LM: 2.0 (0x3); Subversión del nivel LM: 0x430e

El retardo medido en el sentido de transmisión maestro-esclavo, sin configurar calidad de servicio, y empleando paquetes DH del modo básico y EDR, no sólo no responde al comportamiento esperado sino que se caracteriza por una gran inestabilidad y dispersión, tal y como se observa en la figura 5.55. En el sentido esclavo-maestro el comportamiento es incluso peor, por lo que no se ha considerado relevante incluir los resultados.

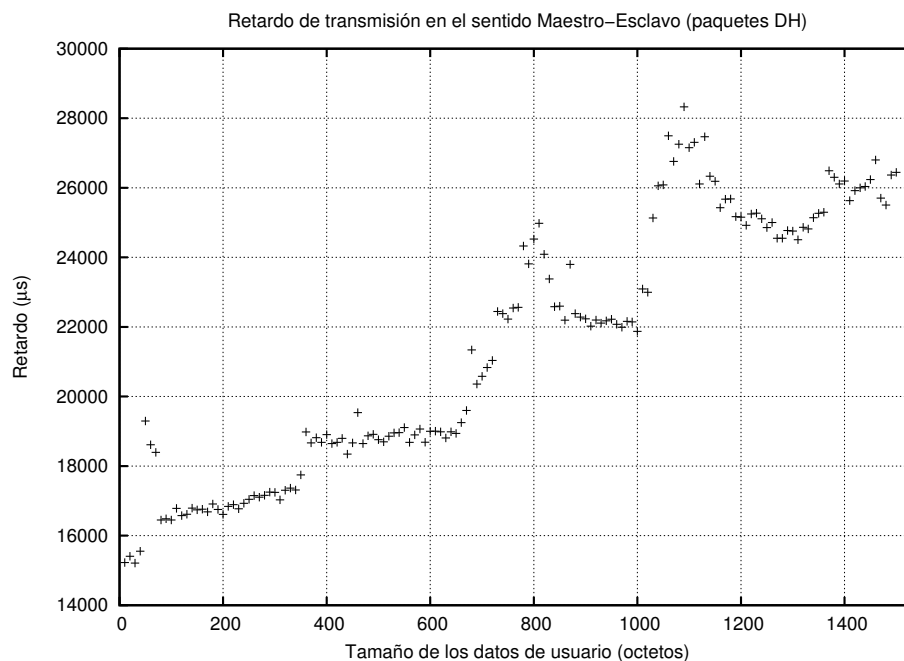
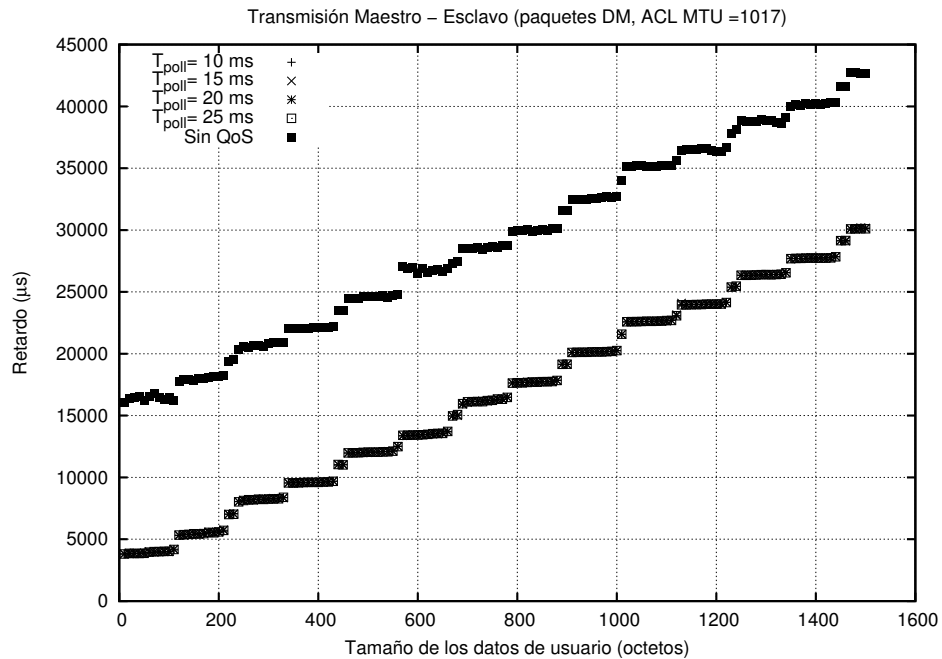


Figura 5.55.: Representación del retardo medio de transmisión sin configurar calidad de servicio, medido con dispositivos *Broadcom*, para paquetes DH y valor del parámetro ACL MTU de 1017 octetos.

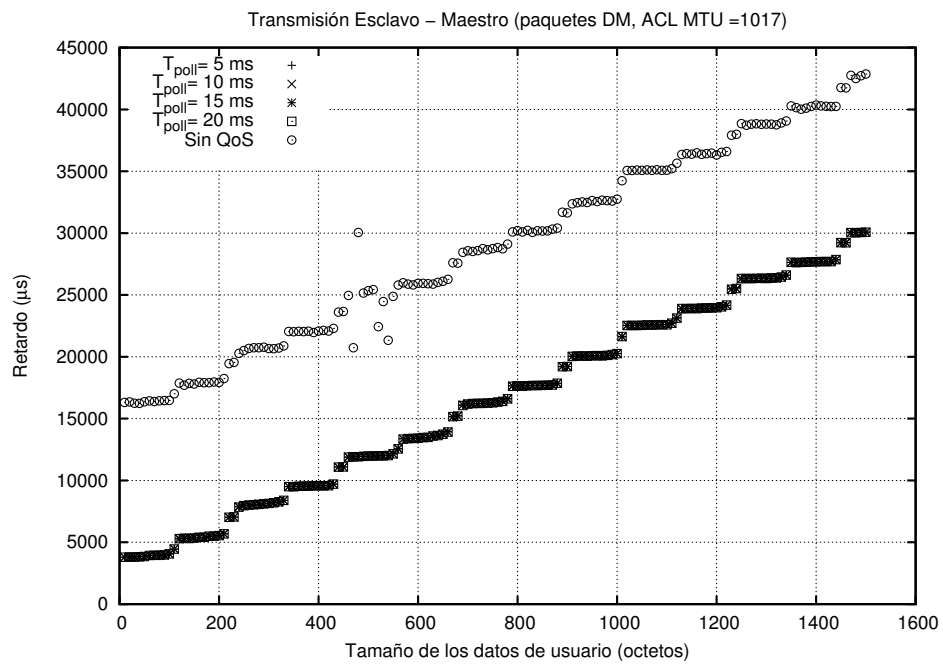
Por contra, cuando se configuran los dispositivos para utilizar exclusivamente paquetes DM, los retardos medidos, a diferencia de lo que sucede con dispositivos ISS, responden al modelo teórico, aunque no al básico, sí al modificado que se presenta en el apartado 5.3.1.4 de [Moron08] en el que se redefine la componente $t_{TX}(x)$, de acuerdo con la ecuación 5.40, suponiendo que el retardo no depende exactamente del número de bits a transmitir sino del número de *slots* ocupados por el paquete que se utiliza:

$$t_{TX_{Broadcom}}^{DM}(x) = \begin{cases} 0 & x = 0 \\ T_S & 0 < x \leq L_1 \\ 3 \cdot T_S & L_1 < x \leq L_3 \\ 5 \cdot T_S & L_3 < x \leq L_5 \\ t_{ACK}(L_5) \cdot \left\lfloor \frac{x}{L_5} \right\rfloor + & x > L_5 \\ + t_{TX}(x \bmod L_5) & \end{cases} \quad (5.40)$$

Con este matiz, el retardo teórico coincide con el medido en ambos sentidos de transmisión considerando un *offset* de 4 *slots* para todos los casos en los que se configura la calidad de servicio, independientemente del valor indicado para el parámetro *Latency*, y de 24 *slots* en caso de que no se configure (véase la figura 5.57). Finalmente, es conveniente aclarar que en esta ocasión tampoco ha sido posible identificar el efecto de la capa de transporte debido a que el dispositivo no soporta el comando *HCI_Write_Loopback_Mode*.

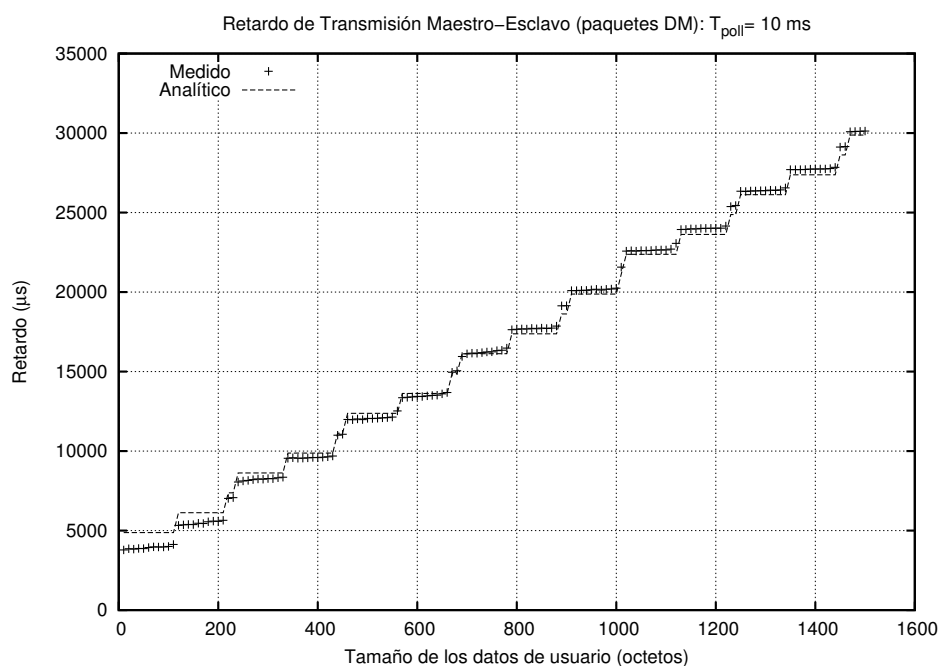


(a) Sentido de transmisión maestro-esclavo

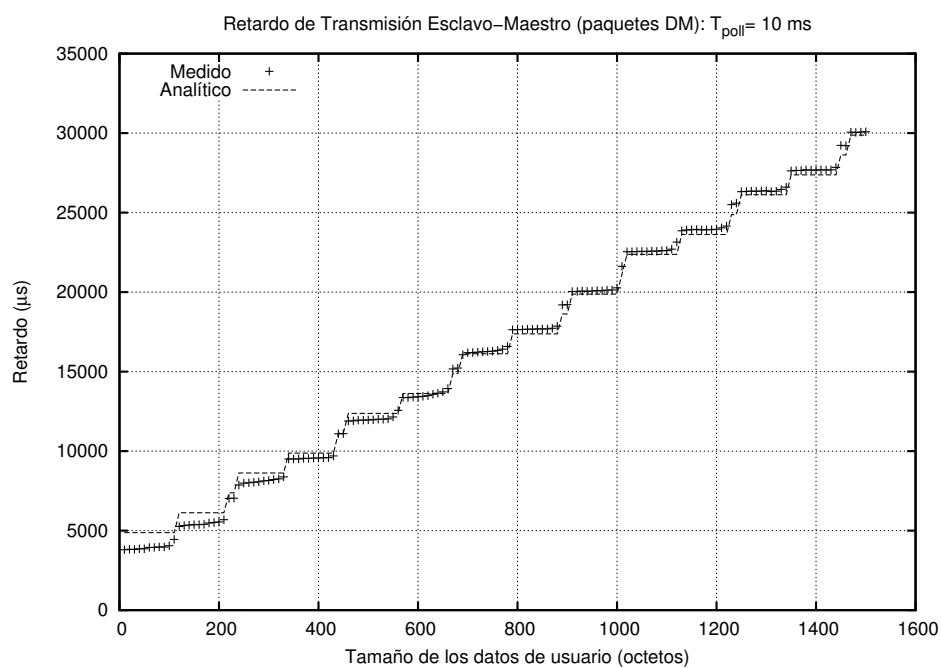


(b) Sentido de transmisión esclavo-maestro

Figura 5.56.: Efecto de la configuración de la calidad de servicio en dispositivos *Broadcom*, para paquetes DM y un valor del parámetro ACL MTU de 1017 octetos.



(a) Sentido de transmisión maestro-esclavo



(b) Sentido de transmisión esclavo-maestro

Figura 5.57.: Comparación del retardo medio de transmisión teórico con el medido, utilizando paquetes DM, para dispositivos *Broadcom*.

CAPÍTULO 6

Evaluación empírica del modelo teórico teniendo en cuenta el efecto de las retransmisiones

En este capítulo se expone cómo se ha llevado a cabo la validación empírica del retardo teórico calculado, en el apartado 4.4, para el perfil SPP teniendo en cuenta el efecto de las retransmisiones. Para ello, tras describir en la sección 6.1 la metodología de medida, en la sección 6.2, se analizan los resultados de las pruebas realizadas, comenzando, en el apartado 6.2.1, por detallar el procedimiento seguido para la obtención de la tasa BER, la cual es comparada, en el apartado 6.2.2, con la prevista teóricamente a partir de la relación SNR. Por último, la sección 6.2 concluye con la comparación del retardo medido con el retardo calculado analíticamente a partir de la tasa BER medida y computada en distintos casos.

6.1. Metodología de medida

Con el fin de evaluar la validez del modelo teórico propuesto considerando el efecto de las retransmisiones, se ha diseñado un escenario que permite reproducir empíricamente

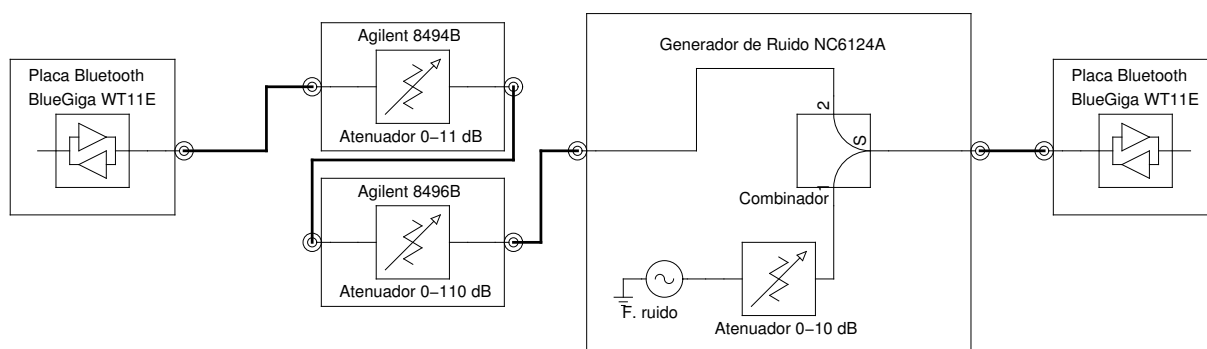


Figura 6.1.: Diagrama de conexión de componentes

el modelo de canal gaussiano propuesto teóricamente, en el que se considera que las transmisiones entre dos dispositivos Bluetooth sólo se ven afectadas por un factor de atenuación y la adición de un nivel de ruido blanco gaussiano. Para ello, se ha establecido un camino conducido entre los transceptores de los dos dispositivos, conectándolos, mediante cable coaxial, a un generador de ruido blanco y a un atenuador de acuerdo con el esquema de la figura 6.1.

En los siguientes epígrafes se describen las características de cada uno de los componentes del sistema de medida utilizado, presentado en la figura 6.2.



Figura 6.2.: Vista general del sistema de medida

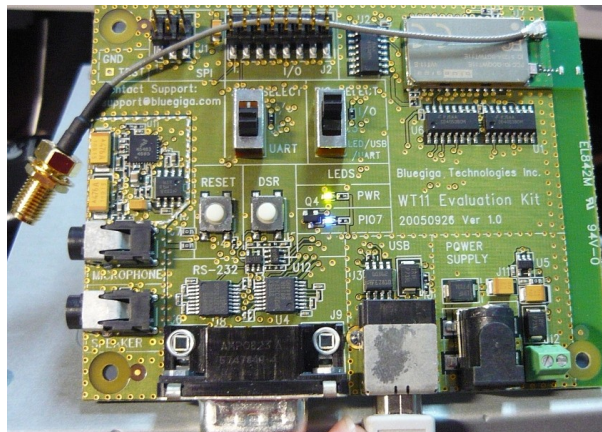


Figura 6.3.: Placas Bluetooth de desarrollo BlueGiga WT11-E

Placas de desarrollo Bluetooth

Los dispositivos comerciales empleados como unidades Bluetooth en todas las pruebas presentadas en el capítulo 5 han sido reemplazados por placas de desarrollo, del fabricante BlueGiga[BlueGiga Tech.a], que incorporan el mismo *chipset* de fabricante CSR, *BlueCore* 4. Estas placas están fabricadas con el objetivo específico de servir de plataforma de pruebas para otros fabricantes que quieran desarrollar dispositivos que integren la tecnología Bluetooth. Con este fin, constan de un módulo Bluetooth montado sobre una placa especial que proporciona múltiples elementos de conectividad, alimentación, etc. Concretamente, las placas utilizadas forman parte del *Kit* de evaluación EKWT11-E [BlueGiga Tech.b], integrado por dos placas y una interfaz de programación SPI (*Serial Peripheral Interface*) para puerto paralelo estándar de PC. Cada placa, mostrada en la figura 6.3, consta de un módulo Bluetooth WT11-E que se caracteriza por:

- Estar basado en el *chipset* *BlueCore* 4.
- Ser de clase 1, lo que significa que puede transmitir con una potencia superior a 4 dBm, hasta un máximo 12 dBm.
- Integrar una memoria *flash* externa para permitir la actualización del *firmware*.
- Disponer de un conector de antena externa de tipo U.FL y de un cable adaptador (*pig-tail*) a conector SMA.

Como ya se ha indicado, el módulo Bluetooth WT11-E se encuentra instalado sobre una placa de desarrollo que proporciona:

- Alimentación al módulo, ya sea desde fuentes de alimentación de 12 V externas o desde la interfaz USB.
- Acceso a la interfaz de programación SPI.
- Distintas interfaces de comunicaciones con el módulo, bien a través de puerto serie RS-232, o bien a través de la interfaz USB.
- Conexión con elementos externos, que permiten acceder a las entradas y salidas digitales del módulo.

Los módulos vienen cargados con un *firmware* compatible con la especificación Bluetooth 2.0 + EDR, específico del fabricante BlueGiga, y que permite su control por puerto serie utilizando un protocolo basado en comandos AT. Sin embargo, dado que se dispone de la interfaz SPI y de herramientas *software* para configurar el módulo, es posible sustituir el *firmware* precargado por otras versiones tanto personalizadas por el fabricante BlueGiga, como por versiones estándar (aunque para desarrollo), del fabricante del *chipset* CSR, compatibles con las versiones 1.2, 2.0 + EDR y 2.1 + EDR de Bluetooth. Finalmente se ha seleccionado, para las pruebas presentadas en este capítulo, un *firmware* estándar del fabricante del *chipset* CSR para desarrollo, compatible con la versión Bluetooth 2.0 + EDR. En concreto, la versión es:

- Versión de HCI: 2.0 (0x3); Revisión de HCI: 0xafa
- Versión del nivel LM: 2.0 (0x3); Subversión del nivel LM: 0xafa

Atenuador

Para simular la atenuación introducida por la propagación radio entre el transmisor y el receptor se han utilizado atenuadores ajustables para RF. Con el fin de poder garantizar el funcionamiento correcto en la banda ISM de 2,4 GHz, se han seleccionado atenuadores ajustables del fabricante Agilent que están diseñados para proporcionar un valor de

atenuación calibrado desde continua hasta 18 GHz. Los modelos elegidos son 8494B [Agilent Tech.a] y 8496B [Agilent Tech.b], que permiten introducir, respectivamente, un nivel de atenuación en un rango de 0 a 11 dB, en pasos de 1 dB, y de 0 a 110 dB, en pasos de 10 dB. Los atenuadores se han conectado en serie, tal y como se ilustra en la figura 6.4a.

Generador de ruido

El segundo elemento necesario para modelar un canal gaussiano es introducir un determinado nivel de ruido blanco, es decir, de potencia de señal interferente uniforme en toda la banda considerada. Para ello se ha empleado un generador de ruido del fabricante NoiseCom [NoiseCom]. En concreto, el modelo utilizado es el NC6124A, mostrado en la figura 6.4b, que, según especificaciones, emite -10 dBm en la banda de 2 a 4 GHz.

6.2. Análisis de los resultados

Al igual que en [Moron08] la evaluación del efecto de las retransmisiones en el retardo requiere expresar el número medio de retransmisiones en función de la tasa PER, calculada a partir de la tasa de bits erróneos BER. Además, el escenario empleado para la validación empírica del modelo, permite plantearse el reto de verificar si la tasa BER medida coincide con la obtenida teóricamente a partir de la relación SNR, mediante las ecuaciones 4.30, para GFSK, y 4.35 evaluada con $M = 4$ y $M = 8$, para el caso de $\pi/4$ DQPSK y 8 DPSK, respectivamente. Por consiguiente, para exponer los resultados, en el apartado 6.2.1 se va a comenzar describiendo la medición de la tasa BER. A continuación, en el apartado 6.2.2, se prosigue comparando la tasa BER medida con la estimada a partir de la relación SNR, y se finaliza, en el apartado 6.2.3, comparando el retardo medido en las condiciones descritas en el apartado 6.1 con el retardo calculado analíticamente a partir de la tasa BER medida y computada para distintas combinaciones de valores de nivel de ruido y atenuación.



(a) Atenuadores calibrados ajustables Agilent 8494B y 8496B (fuente, [Agilent Tech.a, Agilent Tech.b])



(b) Generador de ruido NoiseCom NC6124A (fuente: [NoiseCom])

Figura 6.4.: Instrumentación utilizada para modelar el comportamiento de un canal gaussiano

6.2.1. Medición de la tasa BER

Tanto en el entorno de medida presentado en [Moron08], como en el utilizado para las medidas que se presentan en esta sección, no es factible imponer la tasa BER directamente. Por esto motivo, se ha optado por inducirla, mediante la introducción de ruido, y estimar la tasa BER resultante. Así, del mismo modo que en [Moron08],

para estimar la tasa BER conseguida para cierto nivel de ruido, en una primera aproximación, se ha utilizado el comando *HCI_Read_Link_Quality*¹ (véase el apartado 7.5.3 de [Bluetooth SIG04, Bluetooth SIG07, vol. 2, parte E]) que proporciona el parámetro *LQ*, *Link Quality*. Este parámetro, un entero entre 0 y 255, se relaciona con la tasa BER tal y como especifica el fabricante [CSR], según la siguiente ecuación:

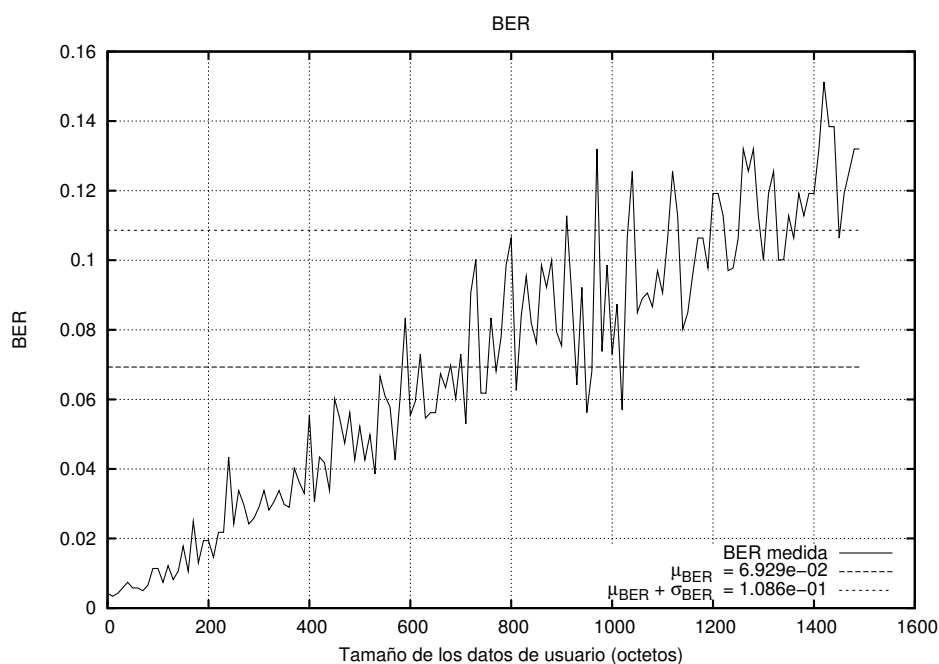
$$BER = \begin{cases} (255 - LQ) \cdot 0,25 \cdot 10^{-4} & 215 \leq LQ \leq 255 \\ 0,001 + (214 - LQ) \cdot 8 \cdot 10^{-4} & 90 \leq LQ \leq 215 \\ 0,1 + (89 - LQ) \cdot 64 \cdot 10^{-4} & 0 \leq LQ \leq 90 \end{cases} \quad (6.1)$$

Sin embargo, los resultados de pruebas preliminares ponen de manifiesto la falta de precisión de los valores de BER obtenidos a partir de esta ecuación. Como ejemplo, cabe mencionar que, para el mismo valor de atenuación del camino de transmisión (45 dB) y de atenuación de ruido (4 dB), los dispositivos proporcionan para el parámetro *LQ* valores muy dispares dependiendo de que actúen como esclavo o como maestro, según se ilustra en las gráficas de la figura 6.5, en la que se representan los valores de la tasa BER en las citadas condiciones, para tamaños de datos de usuario comprendidos entre los 10 y los 1500 octetos. Además, en la figura 6.5b, correspondiente al sentido de transmisión maestro-esclavo, se observa otra evidencia de la inexactitud de la tasa BER obtenida: el orden de magnitud del valor promedio de dicho parámetro, 10^{-1} , valor que, de acuerdo con las especificaciones del fabricante representa el límite a partir del cual es bastante improbable que se mantenga la conexión.

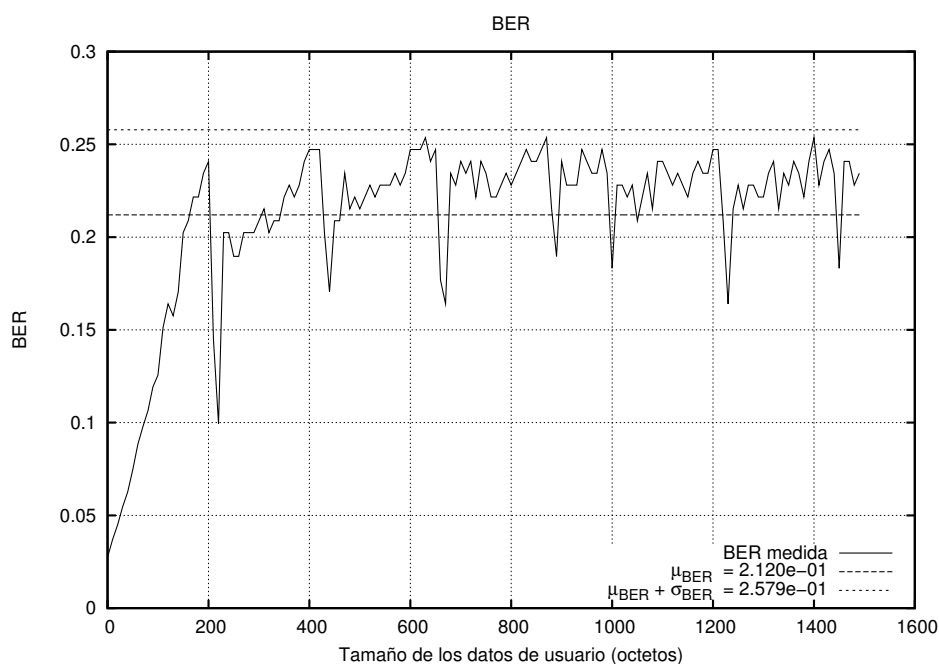
Como alternativa a este método de medición de la BER se ha optado por utilizar las capacidades de prueba de la capa RF implementadas en los módulos Bluetooth, bajo la designación de «Modo de Test» [Bluetooth SIG04, Bluetooth SIG07, vol. 3, parte D]. En este modo de prueba, se configura al subsistema RF del módulo con el fin de medir ciertos parámetros de calidad y rendimiento definidos en la especificación para este nivel.

Concretamente, se ha empleado la aplicación *BlueTest*, aplicación específica proporcionada por el fabricante CSR, cuya interfaz gráfica se presenta en la figura 6.6, que permite configurar este modo de test en módulos que incorporen su *chipset*. La aplicación *BlueTest*

¹Comando equivalente a *HCI_Get_Link_Quality* definido para Bluetooth 1.1



(a) Sentido de transmisión maestro-esclavo



(b) Sentido de transmisión esclavo-maestro

Figura 6.5.: Tasa BER obtenida, a partir de la ecuación 6.1, para un nivel de atenuación de 45 dB y 4 dB de ruido en los sentidos de transmisión maestro-esclavo y esclavo-maestro.

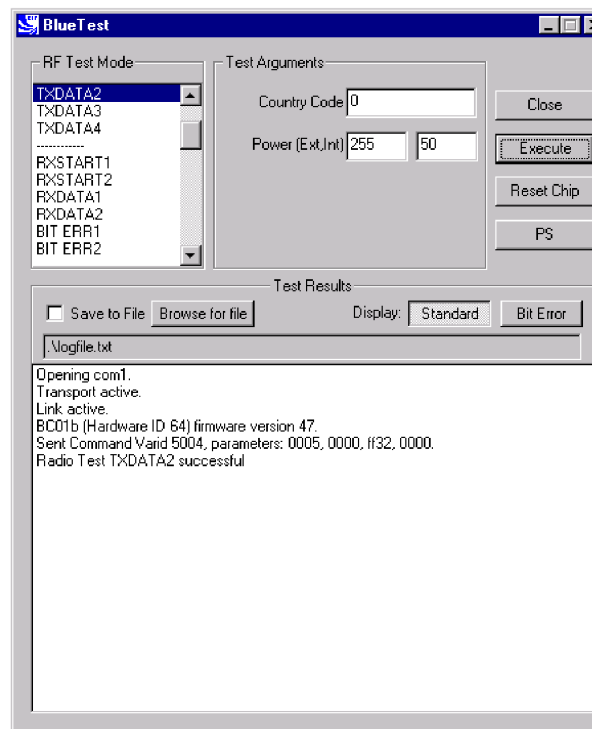


Figura 6.6.: Aplicación *BlueTest* del fabricante CSR

se comunica con los módulos utilizando protocolos y comandos específicos del fabricante, mediante una de las interfaces de comunicación de las que disponen los módulos (USB, puerto serie RS-232, SPI). Desde esta aplicación es posible configurar parámetros generales de las pruebas, como son: el tipo de paquete, el número de octetos de información a enviar en cada paquete, el intervalo entre transmisiones consecutivas, etc. Además, también permite configurar parámetros específicos de cada prueba como: el canal de transmisión, la secuencia de salto, la potencia de transmisión, el número de muestras promediadas para medidas de BER, el periodo entre obtención de informes, etc.

El procedimiento seguido para obtener la tasa BER consta de los siguientes pasos:

1. Se han configurado ambos módulos, el transmisor y el receptor, para que compartan los mismos parámetros generales, que son los siguientes:
 - Tipo de Paquete: DH5.
 - Número de octetos del campo de datos: 339 octetos.

- Intervalo entre transmisiones: 5000 μ s.
2. En el módulo transmisor se ha seleccionado una prueba específica de transmisión radio, designada por el fabricante CSR como TXDATA2, que se configura para que transmita una secuencia de paquetes, del tipo y tamaño seleccionados en los parámetros generales, siguiendo una secuencia reducida de salto en toda banda de transmisión de Bluetooth.

En esta prueba, según se especifica, los paquetes contienen una secuencia de bits pseudoaleatoria, denominada PRBS9 (*Pseudo-Random Bit Sequence 9*). Esta secuencia estándar, especificada en [ITU-T96] y referenciada en [Bluetooth SIG04, Bluetooth SIG07, vol. 2, parte A], se utiliza para la realización de pruebas de rendimiento a equipos de transmisión digital y se genera mediante un registro de desplazamiento realimentado de longitud 9, generando una secuencia con periodo de 511 bits.

3. Se ha configurado la potencia del transmisor, ajustando los parámetros de ganancia del amplificador interno del *chipset* y del amplificador externo para transmitir a máxima potencia, 12 dBm, según especificaciones [Bluetooth SIG04, Bluetooth SIG07]. Los valores de los parámetros de ganancia de los amplificadores que han de configurarse para lograr este nivel de potencia —255 (interno) y 30 (externo)— se han obtenido de la tabla de calibración de potencia, almacenada en memoria no volátil de configuración del *chipset* (*Persistent Storage*), utilizando para ello la herramienta *software* de fabricante *PSTools*.
4. En el módulo receptor se ha seleccionado la prueba de recepción radio denominada por el fabricante CSR como BIT_ERR2, que permite obtener estimaciones de la tasa BER. En esta prueba se sigue la misma secuencia reducida de salto para recibir paquetes de información y extraer la información contenida. El módulo realiza todas las operaciones de recepción y compara los paquetes recibidos con la información esperada, considerando que transportan la secuencia de bits PRBS9. De esta forma es posible estimar la BER residual, es decir, una vez aplicados los mecanismos de corrección de errores presentes en la codificación de un paquete Bluetooth. La información de BER se calcula y recibe periódicamente, según se ha configurado en los parámetros generales.

Al respecto es conveniente aclarar que, precisamente porque la tasa BER que se obtiene como resultado de la prueba BIT_ERR2, es una tasa residual, se ha elegido un paquete DH5 en lugar de un paquete DM5, dado que la información transportada por los paquetes DH no está protegida por ningún código corrector y, por tanto, la BER obtenida (residual) se aproxima en gran medida a la real.

Los valores de la tasa BER medidos, mediante el procedimiento especificado, para una atenuación de 45 dB y distintos niveles de ruido se indican en la tabla 6.1, para las modulaciones GFSK, $\pi/4$ DQPSK y 8 DQPSK. Nótese que, para estos dos últimos casos, hay niveles de atenuación para los que no se proporciona la tasa BER. La razón no es otra que la imposibilidad de haber obtenido un valor estable.

Es preciso aclarar que el motivo por el que se ha elegido el nivel de atenuación de 45 dB para los atenuadores calibrados está relacionado con el rango óptimo de recepción y se detalla en el apartado 6.2.2.2.

Seguidamente, en el apartado 6.2.2, los valores medidos se comparan con los calculados teóricamente según se expone en la sección 4.4.1.

6.2.2. Comparación de la tasa BER medida con la obtenida teóricamente a partir de la relación SNR

Se ha utilizado el analizador de espectro de Agilent Technologies modelo N9010A [Agilent Tech.c], mostrado en la figura 6.7, para medir la potencias de señal y ruido reales, según se describe en los apartados 6.2.2.1 y 6.2.2.2. A continuación, se determina la relación SNR (apartado 6.2.2.3), y se compara la tasa BER correspondiente a dicha SNR (obtenida a partir de las ecuaciones especificadas en el apartado 4.4.1), con la tasa BER medida según el procedimiento expuesto en el apartado 6.2.1 (apartado 6.2.2.4).

6.2.2.1. Medida de la potencia real de ruido

El procedimiento que se ha seguido se detalla a continuación:

Tabla 6.1.: Resultados de las medidas de BER obtenidos para las modulaciones GFSK, $\pi/4$ DQPSK y 8 DQPSK, con una atenuación de señal de 45 dB.

Atenuación de ruido (dB)	BER GFSK	BER $\pi/4$ DQPSK	BER 8 DQPSK
0	$4,24 \cdot 10^{-2}$	$4,84 \cdot 10^{-2}$	—
1	$2,67 \cdot 10^{-2}$	$3,14 \cdot 10^{-2}$	—
2	$1,49 \cdot 10^{-2}$	$1,50 \cdot 10^{-2}$	—
3	$9,22 \cdot 10^{-3}$	$8,00 \cdot 10^{-3}$	$8,70 \cdot 10^{-2}$
4	$5 \cdot 10^{-3}$	$4,28 \cdot 10^{-3}$	$6,70 \cdot 10^{-2}$
5	$2,3 \cdot 10^{-3}$	$1,53 \cdot 10^{-3}$	$4,20 \cdot 10^{-2}$
6	$1,1 \cdot 10^{-3}$	$6,90 \cdot 10^{-4}$	$3,10 \cdot 10^{-2}$
7	$4,8 \cdot 10^{-4}$	$1,84 \cdot 10^{-4}$	$2,14 \cdot 10^{-2}$
8	$1,3 \cdot 10^{-4}$	$6,10 \cdot 10^{-5}$	$1,24 \cdot 10^{-2}$
9	$5,7 \cdot 10^{-5}$	—	$8,20 \cdot 10^{-3}$
10	$1,7 \cdot 10^{-5}$	—	$4,7 \cdot 10^{-3}$

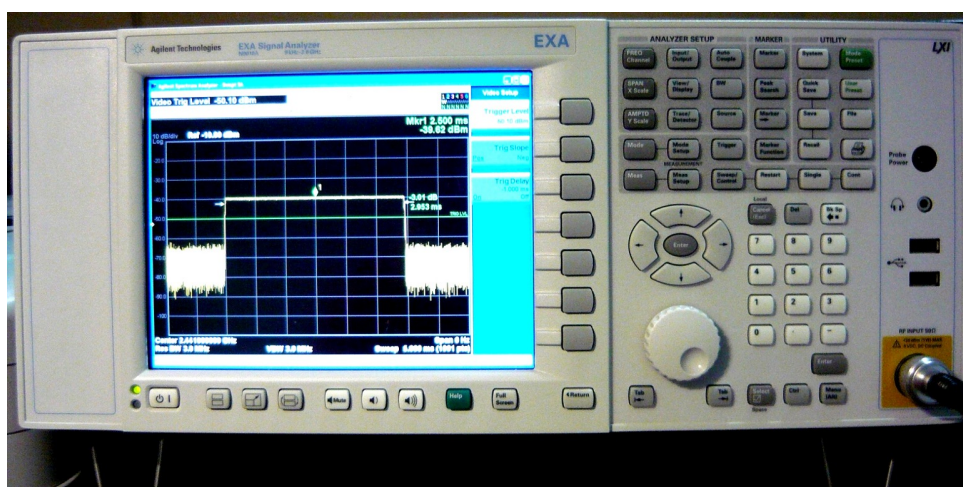


Figura 6.7.: Analizador de espectro Agilent Technologies N9010A.

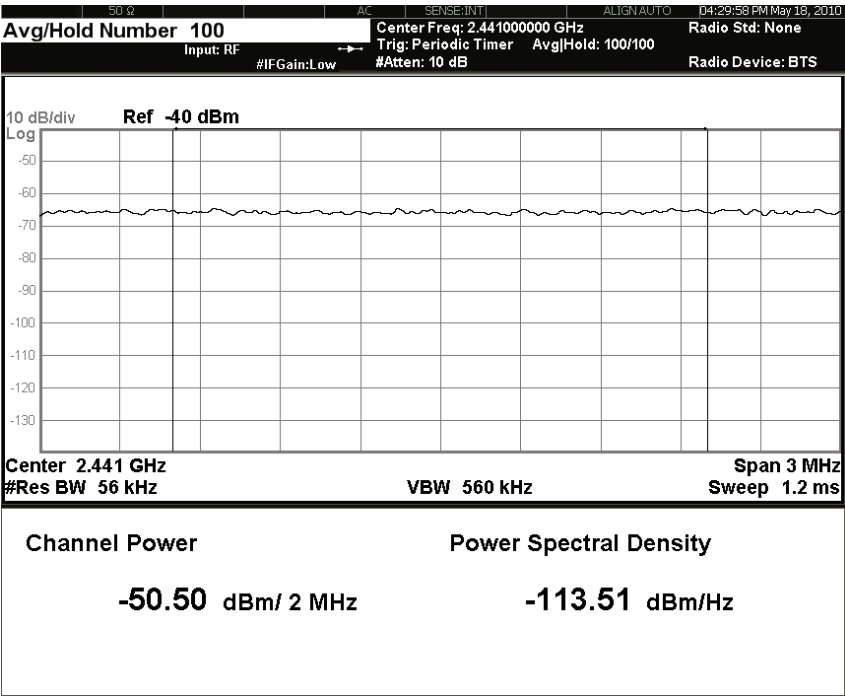
1. Se ha conectado la salida del generador de ruido a la entrada del analizador de espectro utilizando un cable con conector SMA, en el extremo del generador, y conector F para el analizador.

2. Se ha configurado el analizador de espectro para realizar medidas de potencia en el canal de transmisión (*Channel Power*).
3. Se ha sintonizado el centro de la banda a 2441 MHz.
4. Se ha ajustado el ancho de banda de resolución a 62 KHz, al igual que se ha hecho para medir la potencia de señal, con lo que se garantiza un tiempo de barrido inferior a 1 ms.
5. En torno al centro de la banda (2441 MHz) se han realizado dos medidas, promediando para 100 repeticiones:
 - a) Ancho de banda de medida (BW, *bandwidth*) de 2 MHz, equivalente al ancho de banda de transmisión de la señal Bluetooth.
 - b) Ancho de banda de medida de 80 MHz, para promediar el ruido en toda la banda de transmisión.

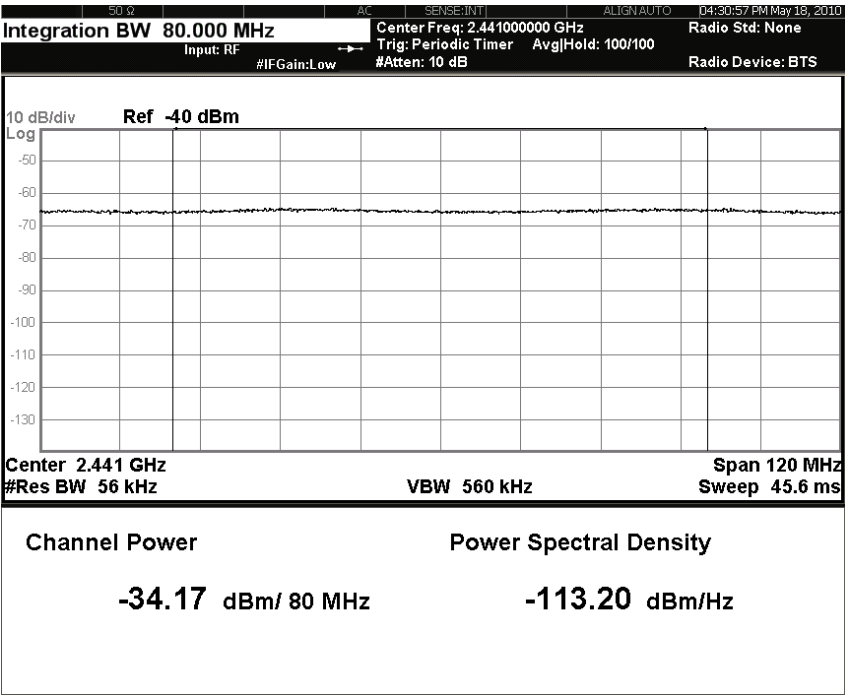
En la tabla 6.2 se indica la potencia de ruido para ambas medidas y para cada uno de los niveles de atenuación indicados en la primera columna. Además, a modo de ejemplo, en la figura 6.8, se muestran dos capturas de pantalla para el caso en el que el nivel de atenuación de ruido se ha fijado a 4 dB.

A partir de las medidas realizadas, puede concluirse que:

- La diferencia de ruido promedio en toda la banda frente a la obtenida en el canal situado en el centro de la banda (2441 MHz) resulta ser inferior a 1 dB, pudiéndose garantizar así que el generador introduce ruido blanco en toda la banda considerada.
- El generador de ruido, a su salida, tras el combinador, emite con una densidad espectral de potencia de $-109,1$ dBm/Hz, atenuables de 0 dB a 10 dB, pudiéndose reducir la potencia de ruido hasta $-119,1$ dBm/Hz.



(a)



(b)

Figura 6.8.: Capturas de la vista del analizador de espectro en la que se muestra las potencias de ruido medidas para el nivel de atenuación de 4 dB y para un ancho de banda de 2 MHz y de 80 MHz, en torno a la frecuencia central de 2441 MHz.

Tabla 6.2.: Potencias de salida reales medidas para el rango de atenuación comprendido entre 0 dB y 10 dB y para un ancho de banda de medida (BW, *bandwidth*) de 2 MHz y de 80 MHz, en torno a la frecuencia central de 2441 MHz

Nivel de atenuación (dB)	BW = 2 MHz (dBm/Hz)	BW = 80 MHz (dBm/Hz)
0	-109,12	-109,18
1	-110,33	-110,26
2	-111,44	-111,36
3	-112,32	-112,20
4	-113,27	-113,15
5	-114,44	-114,26
6	-115,22	-115,12
7	-116,15	-116,14
8	-117,48	-117,40
9	-118,09	-118,08
10	-119,11	-119,10

6.2.2.2. Medida de la potencia real de señal

La medición de la potencia real de señal, en el caso del modo básico de Bluetooth, se ha llevado a cabo mediante el siguiente procedimiento:

1. Se ha utilizado la aplicación *BlueTest*, para, mediante el modo de test, configurar el módulo transmisor como se indica a continuación:
 - a) Se han configurado los siguientes parámetros generales:
 - Tipo de Paquete: DH5.
 - Número de octetos del campo de datos: 339 octetos.
 - Intervalo entre transmisiones: 5000 μ s.

- b) Se ha seleccionado una prueba específica de transmisión radio, designada por el fabricante CSR como TXDATA1, que es equivalente a la prueba TXDATA2, descrita en el apartado 6.2.1, salvo que en este caso se transmite siempre a una frecuencia fija. Concretamente, se ha sintonizado el transceptor a la frecuencia correspondiente al centro de la banda: 2441 MHz.
 - c) Se ha configurado la potencia del transmisor del mismo modo que se ha hecho para medir la tasa BER (véase el apartado 6.2.1), es decir, ajustando los parámetros de ganancia del amplificador interno del *chipset* y del amplificador externo, respectivamente, a 255 y 30.
2. Se ha realizado la medida de potencia de tres modos diferentes, para la frecuencia central:
- a) Conectando directamente el analizador a la salida de la placa, obteniéndose $-53,44 \text{ dBm/Hz}$.
 - b) Intercalando el atenuador calibrado a 45 dB, resultando $-98,11 \text{ dBm/Hz}$.
 - c) Conectando la salida del atenuador a la entrada del generador de ruido, la potencia medida a la salida del combinador ha sido de $-102,60 \text{ dBm/Hz}$ (véase la figura 6.9).
- Nótese que, con la tercera medida se puede caracterizar la atenuación introducida por el combinador: 4,49 dB.
3. Siguiendo el tercer método se ha repetido la medida para otras frecuencias: 2402, 2420, 2460 y 2480 MHz. De los resultados obtenidos para cada una de ellas, además de para la frecuencia central, que se plasman en la tabla 6.3, puede concluirse que la potencia transmitida por el modulo se mantiene constante en toda la banda de operación.

Adicionalmente, se ha medido la potencia de transmisión de forma alternativa, siguiendo los pasos que se indican a continuación:

1. Sintonizando el analizador de espectro en el canal correspondiente al centro de la banda, es decir, a 2441 MHz.

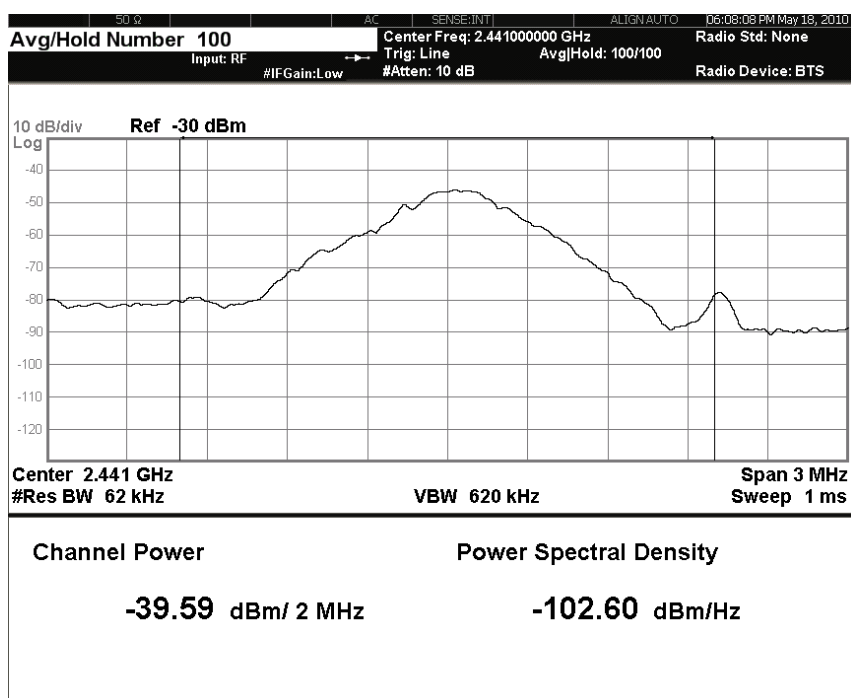


Figura 6.9.: Captura del analizador de espectro en la que se muestra el espectro de potencia correspondiente a la transmisión de un paquete DH5 con 339 octetos, medido a la salida del combinador, a la frecuencia central de 2441 MHz, para el nivel de atenuación de 45 dB y un ancho de banda de 2 MHz.

Tabla 6.3.: Potencias de salida reales medidas para las frecuencias 2402 MHz, 2420 MHz, 2441 MHz, 2460 MHz y 2480 MHz

Frecuencia (MHz)	Potencia de señal (dBm/Hz)
2402	-101,77
2420	-102,53
2441	-102,60
2460	-102,03
2480	-101,73

2. Fijando el *span* a cero.
3. Manteniendo el ancho de banda de resolución a 3 MHz.

4. Configurando el tiempo de barrido a 5 ms.
5. Promediando en potencia (RMS, *Root Mean Square*) 100 medidas.

De esta manera, es posible visualizar en pantalla la variación de la potencia durante la transmisión, obtenida para un paquete DH5 con 339 octetos:

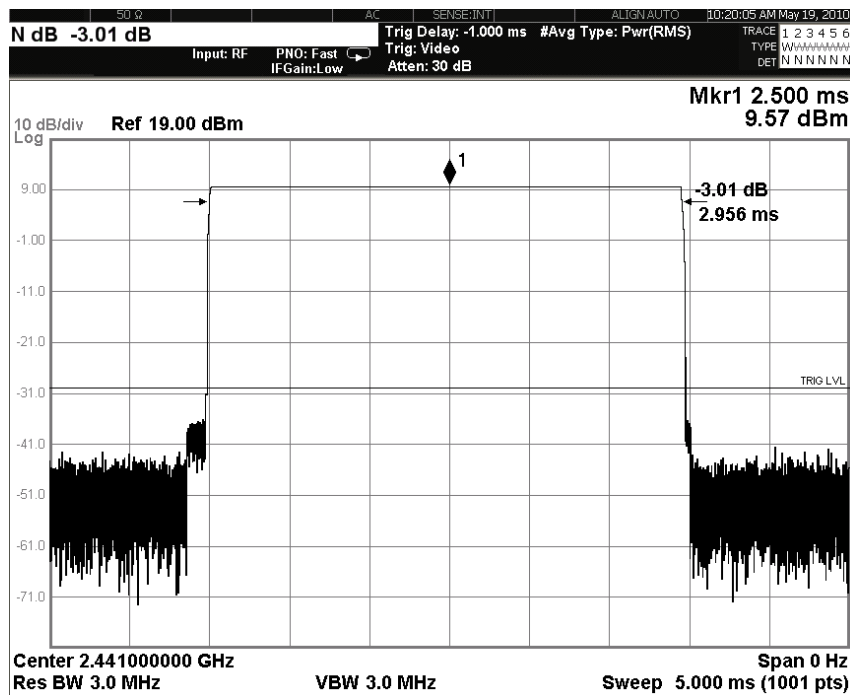
- A la salida del transmisor, es decir, sin atenuador y combinador, según se muestra en la figura 6.10a.
- A la salida del combinador, atenuada de 45 dB, tal y como puede observarse en la figura 6.10b.

Así, se puede constatar que la potencia emitida total es de 9,57 dBm, lo que equivale a los $-53,44$ dBm/Hz medidos en el ancho de banda considerado de 2 MHz por el método 2(a). Análogamente, se puede comprobar que tras el combinador, es decir, a la entrada del receptor se alcanzan $-39,42$ dBm, equivalentes a los $-102,43$ dBm/Hz obtenidos en el ancho de banda de 2 MHz, según el método 2(c) (véase la figura 6.9).

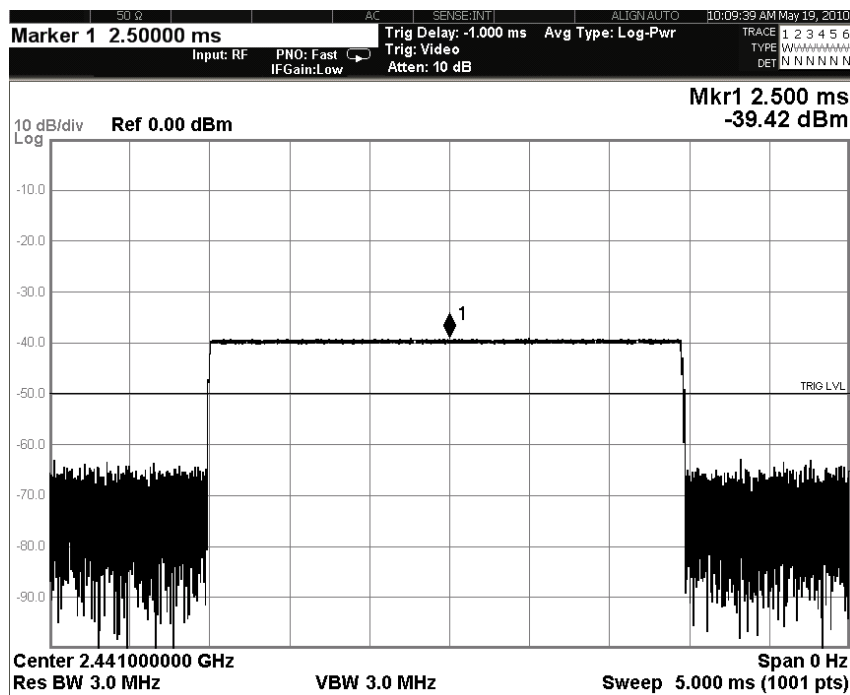
Para el modo EDR de Bluetooth 2.0 + EDR, se ha obtenido, a la salida del combinador, la variación de la potencia de la señal (véase la figura 6.11) y el espectro de potencia (véase la figura 6.12), con la particularidad de que en este caso se ha requerido modificar la potencia del transmisor, ajustando los parámetros de ganancia del amplificador interno del *chipset* y del amplificador externo, respectivamente, a 255 y 80, en vez de a 255 y 30.

Queda por aclarar el motivo por el que se ha seleccionado el valor de 45 dB en los atenuadores calibrados, justificación que revelan los resultados obtenidos de potencia de señal, según se explica a continuación:

En el sistema de medida se establece un camino de transmisión entre el transmisor y el receptor en el que se introducen pérdidas debido a diversos factores, pero cuya principal componente es la atenuación fija establecida por los atenuadores calibrados. Dado que el objetivo que se persigue es medir la tasa BER provocada por la adición de ruido gaussiano, es importante controlar que la señal a la entrada del receptor tenga un valor óptimo para evitar efectos debidos al control de potencia de transmisión o al control de ganancia de recepción. Para ello, hay que tener presente que:

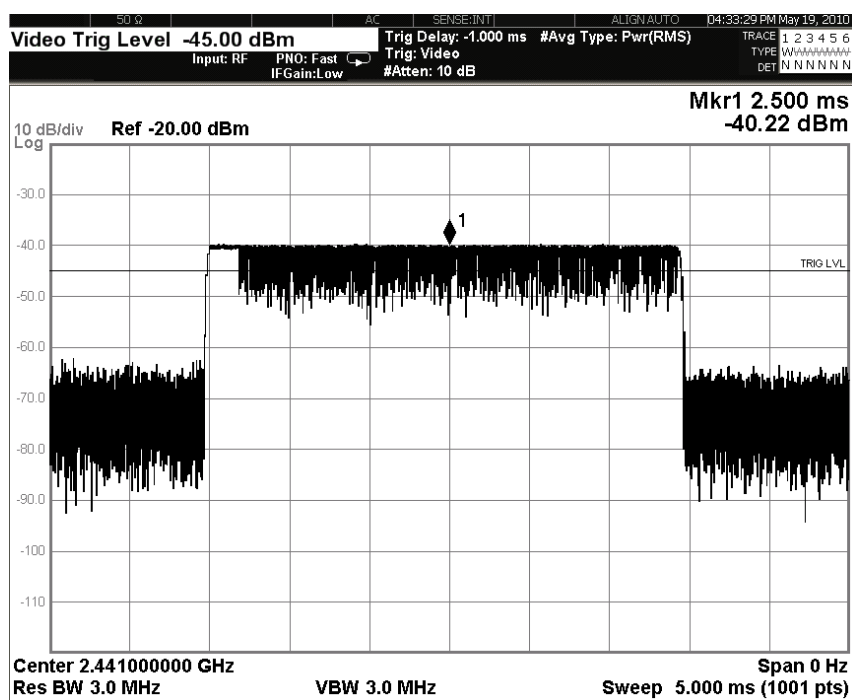


(a) A la salida del transmisor, es decir, sin atenuador y combinador

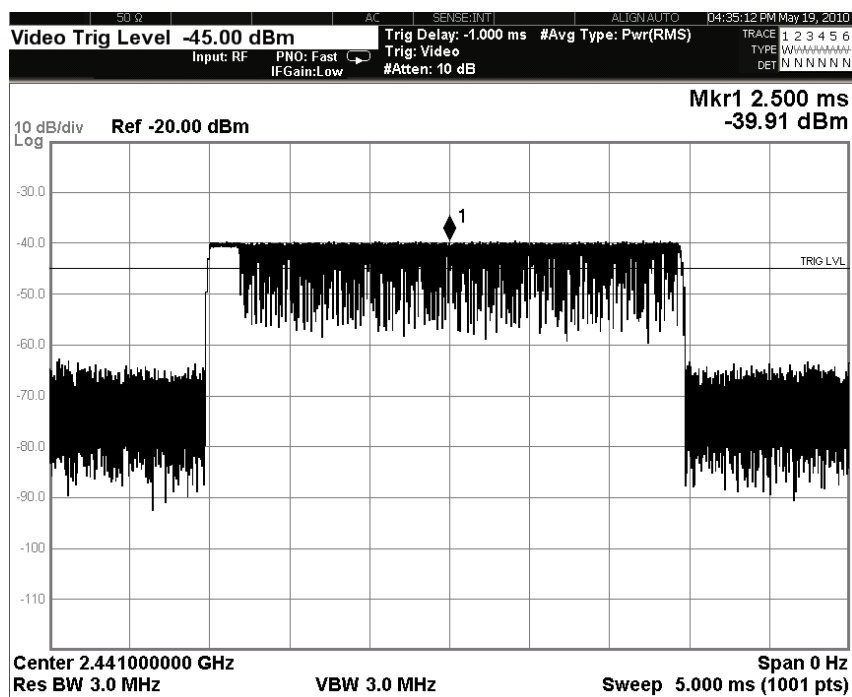


(b) A la salida del combinador, atenuada de 45 dB

Figura 6.10.: Capturas del analizador de espectro en las que se muestra la variación de la potencia durante la transmisión de un paquete DH5 con 339 octetos, obtenidas a la salida del transmisor y del combinador

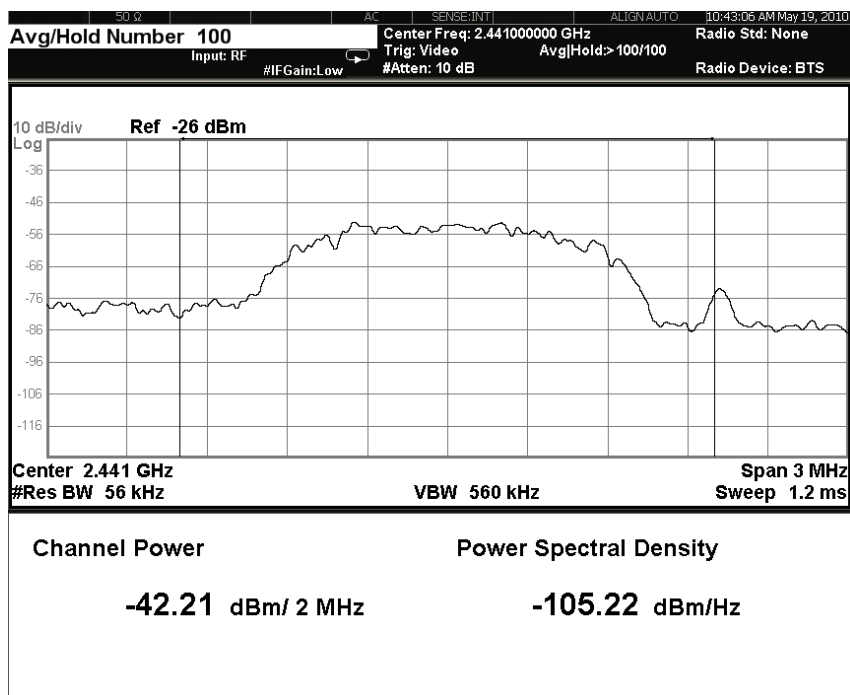


(a) Paquete 2-DH5 con 679 octetos

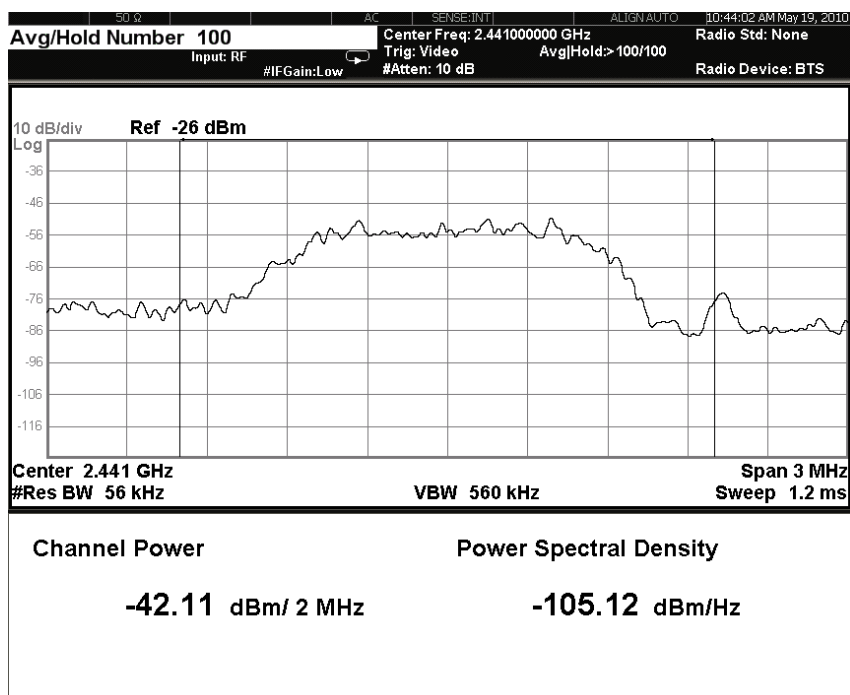


(b) Paquete 3-DH5 con 1021 octetos

Figura 6.11.: Capturas del analizador de espectro en las que se muestra la variación de la potencia durante la transmisión de un paquete 2-DH5 y 3-DH5 con 679 y 1021 octetos



(a) Paquete 2-DH5 con 679 octetos



(b) Paquete 3-DH5 con 1021 octetos

Figura 6.12.: Capturas del analizador de espectro en las que se muestran los espectros de potencia de las transmisiones de un paquete 2-DH5 y 3-DH5 con 679 y 1021 octetos, respectivamente, medidos a la salida del combinador, a la frecuencia central de 2441 MHz, para el nivel de atenuación de 45 dB y un ancho de banda de 2 MHz.

- El control de potencia de transmisión se basa en un mecanismo controlado por el nivel LM, por el que el receptor solicita al transmisor el incremento o decremento de la potencia de salida según el nivel de señal recibida.
- El control de ganancia de recepción opera a nivel RF de forma automática, atenuando o amplificando la señal recibida si esta supera los umbrales máximo o mínimo, respectivamente, que harían operar al receptor fuera de su rango dinámico, degradando, por tanto, su rendimiento.
- El estándar Bluetooth, en el apartado 7.5.4 de [Bluetooth SIG04, Bluetooth SIG07, vol. 2, parte E], especifica el concepto de rango óptimo de señal a la entrada del receptor (*Golden Receive Power Range*). En este sentido, el fabricante CSR, en la documentación del *chipset BlueCore 4*, especifica un rango óptimo mínimo entre -60 dBm y -40 dBm.

Con estas consideraciones, se ha constatado que, al seleccionar un valor de 45 dB en la atenuación introducida en el camino de transmisión, se obtiene un nivel de señal situado en el límite superior del intervalo ($-39,42$ dBm). Así, dicho nivel garantiza que, por una parte, el receptor no utilizará ninguno de los mecanismos anteriores para el ajuste de la señal recibida, y, por otro lado, el nivel de señal será lo suficientemente elevado para que sólo el ruido introducido por el generador tenga influencia en el receptor.

6.2.2.3. Cálculo de la relación SNR a la entrada del receptor

Los resultados de las medidas anteriores permiten calcular la relación SNR a la entrada del receptor. De esta forma se dispone de una referencia absoluta con la que relacionar los datos de tasa BER medidos para los distintos valores de salida atenuada del generador de ruido indicados en la tabla 6.1, con los valores teóricos de BER para cada tipo de modulación de Bluetooth.

En concreto, a partir de la información obtenida de las medidas de potencia de ruido y de señal, se puede afirmar que:

- El nivel de ruido a la entrada del receptor, si no se aplica atenuación en el generador de ruido, es de $-109,1$ dBm/Hz, con una variación inferior a 1 dB, en toda la banda considerada.

- El nivel de señal a la entrada del receptor está comprendido entre $-101,7 \text{ dBm/Hz}$ y $-102,6 \text{ dBm/Hz}$, según el canal medido. Por consiguiente, se va a considerar $-102,15 \text{ dBm/Hz}$ como valor promedio de la banda de transmisión.

En definitiva, a partir de las cantidades anteriores se determina la relación SNR a la entrada del receptor, resultando ser de $6,95 \text{ dB}$ para una atenuación nula del generador de ruido.

6.2.2.4. BER obtenida a partir de la relación SNR medida

Utilizando el valor de relación SNR obtenido a la entrada del receptor, es posible comparar la tasa BER medida con los valores teóricos calculados a partir de las ecuaciones 4.30, para GFSK, y 4.35 evaluada con valores del parámetro M de 4 ($M = 4$) y de 8 ($M = 8$), para el caso de $\pi/4$ DQPSK y 8 DPSK, respectivamente.

En la figura 6.13, se muestran los resultados medidos para la modulación GFSK del modo básico junto con las curvas correspondientes a los índices de modulación $h = 0,28$, $h = 0,30$ y $h = 0,35$. Puede observarse que, para valores de la relación SNR altos, en concreto, aquellos mayores que 14 dB , los valores medidos son claramente superiores a los teóricos. La explicación que, en un principio, cabe argumentar es que la ecuación 4.30 no tiene en cuenta el efecto de la interferencia entre símbolos (ISI) introducida por el filtro gaussiano. En cualquier caso, la tasa BER que corresponde a este nivel de ruido tiene un efecto poco significativo sobre el retardo. De hecho, a partir de las medidas realizadas puede comprobarse que para un nivel de atenuación de ruido de 8 dB , correspondientes a una relación SNR de $14,95 \text{ dB}$, el retardo es prácticamente indistinguible del caso ideal, tal y como se ilustra en la figura 6.14.

Por contra, para valores de SNR inferiores a 10 dB , los valores de la tasa BER medidos están por debajo de la curva correspondiente al mayor índice de modulación ($h = 0,35$). Cabe suponer que, en este caso la tasa BER obtenida experimentalmente infravalora la tasa real debido, probablemente, a que no se computan los paquetes perdidos, que no han sido recibidos a consecuencia del elevado nivel de ruido. No obstante, esta infravaloración no tiene consecuencias para la validación del modelo teórico del retardo, ya que en un escenario de aplicación real el excesivo nivel de ruido acaba provocando la pérdida de la conexión, hecho que impide llevar a término medidas de retardo completas.

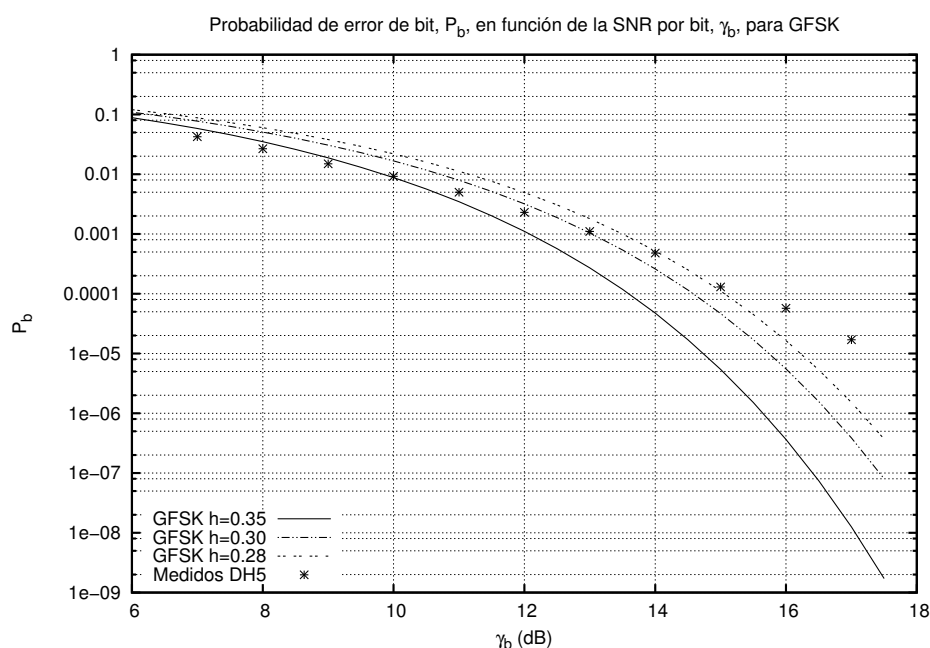


Figura 6.13.: Comparación entre la curva teórica de la probabilidad de error de bit, P_b , para la modulación GFSK, y la tasa BER medida empíricamente para el modo básico de Bluetooth, en función de la relación SNR por bit, γ_b .

En lo que respecta a las modulaciones EDR, $\pi/4$ DQPSK y 8 DPSK, puede observarse, en la figura 6.15, que las tasas BER previstas teóricamente se ajustan en gran medida a las obtenidas de forma empírica, para el rango en el que ha sido factible llevar a cabo la medida, indicado en la tabla 6.1.

6.2.3. Análisis del efecto de las retransmisiones

En este apartado se compara el retardo medido con el previsto teóricamente, incluyendo el retardo introducido para la capa de transporte y considerando como parámetro de entrada:

- La tasa BER medida para los niveles de atenuación de ruido comprendidos entre 4 y 8 dB especificados en la tabla 6.1.
- Las tasas BER computadas por la ecuación 4.30 para los índices de modulación $h = 0,30$ y $h = 0,35$ en el rango de atenuación de ruido indicado.

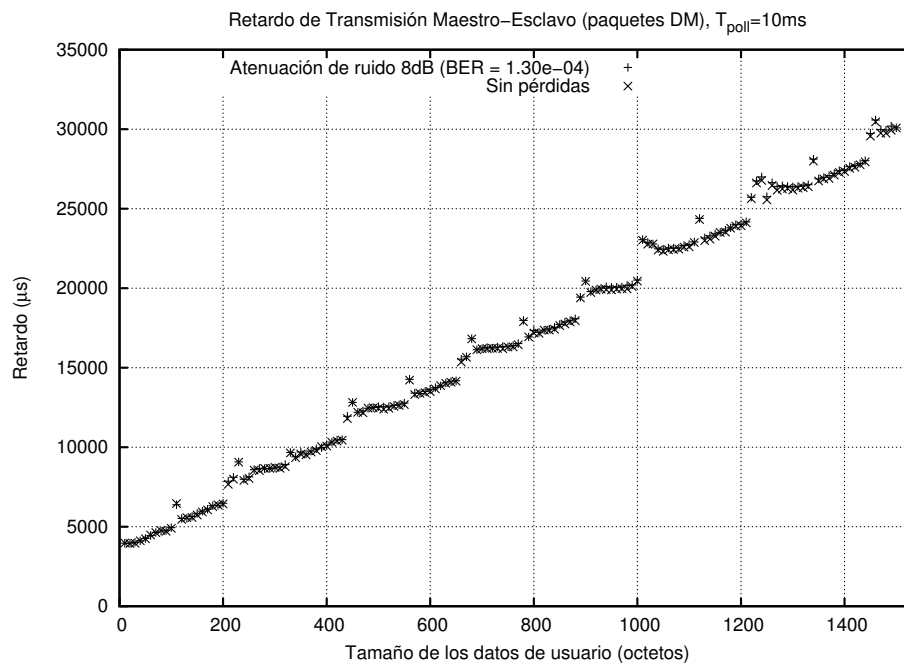
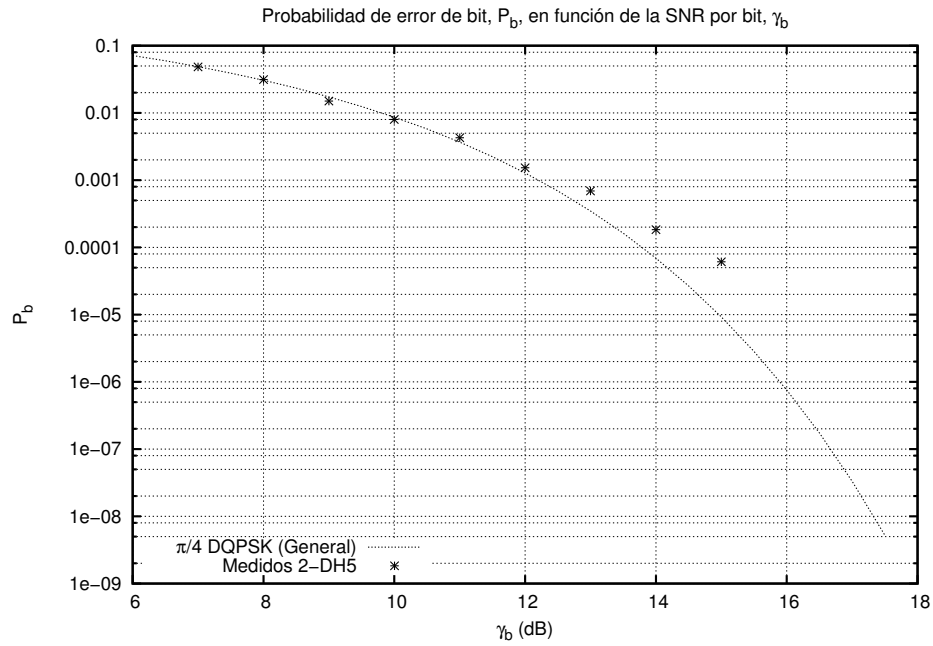


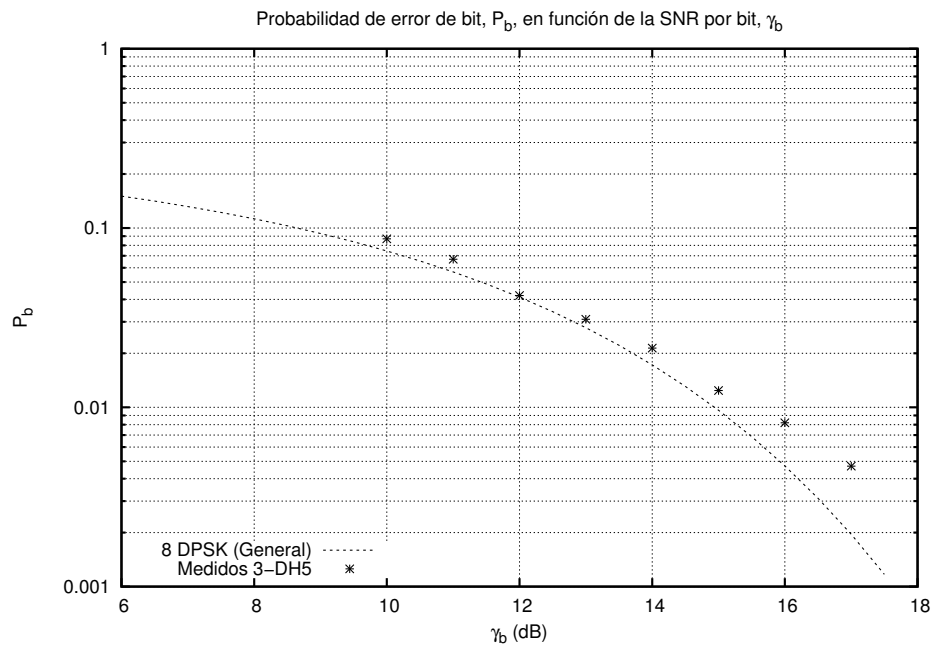
Figura 6.14.: Comparación del retardo medido con una atenuación de ruido de 8 dB (SNR=14,95 dB) y en ausencia de ruido, en el sentido de transmisión maestro-esclavo, con paquetes DM y un intervalo de $poll$ de 10 ms.

Antes de presentar los resultados obtenidos para el caso de paquetes DM, se exponen las conclusiones extraídas de un análisis preliminar de dichos resultados:

- El efecto del intervalo T_{poll} configurado es el mismo que el descrito en el caso en el que no tienen lugar retransmisiones:
 - Sentido esclavo-maestro: El esclavo se ve obligado a esperar que el maestro de la *piconet* lo direcciona mediante el mecanismo de $poll$, de promedio cada $T_{poll}/2$, tiempo al que se suman dos *slots* (véase la ecuación 5.35), incrementándose, por tanto, el retardo de transmisión de datos en esta cantidad.
 - Sentido maestro-esclavo: No existe dependencia del periodo T_{poll} configurado, es decir, el maestro, que gestiona el mecanismo de $poll$, no espera un tiempo variable para direccionar al esclavo, de forma que el retardo adicional es fijo.



(a) Modulación $\pi/4$ DQPSK



(b) Modulación 8 DPSK

Figura 6.15.: Comparación entre la curva teórica de la probabilidad de error de bit, P_b , para las modulaciones EDR, y la tasa BER medida empíricamente para el modo básico de Bluetooth, en función de la relación SNR por bit, γ_b .

Ambos comportamientos se ilustran, a modo de ejemplo, en las figuras 6.16a y 6.16b en ambos sentidos de transmisión y para un nivel de atenuación de ruido de 6 dB, lo que equivale a una tasa BER del 0,11 %.

- El valor del término t_{off} que en el cálculo del retardo de transmisión a nivel de RFCOMM, $\overline{t_R}(x)$, representa un *offset* promedio, ha de ser modificado en ambos sentidos de transmisión:
 - Sentido maestro-esclavo: El intervalo entre direccionamientos consecutivos al esclavo aumenta de los 6 *slots*, medidos con dispositivos de la versión 0xc5c, a 8 *slots*, lo que se traduce en un *offset* promedio de $t_{off} = 4 \cdot T_S$.
 - Sentido esclavo-maestro: Hay que considerar un *offset* adicional que se suma sólo en caso de que se requiera más de un paquete de 5 *slots* para la transmisión de los datos, tal y como se ilustra en la figura 6.17.

Así, el término t_{off} , dado por la ecuación 6.2, ha de ser redefinido para los paquetes DM de acuerdo con la siguiente expresión:

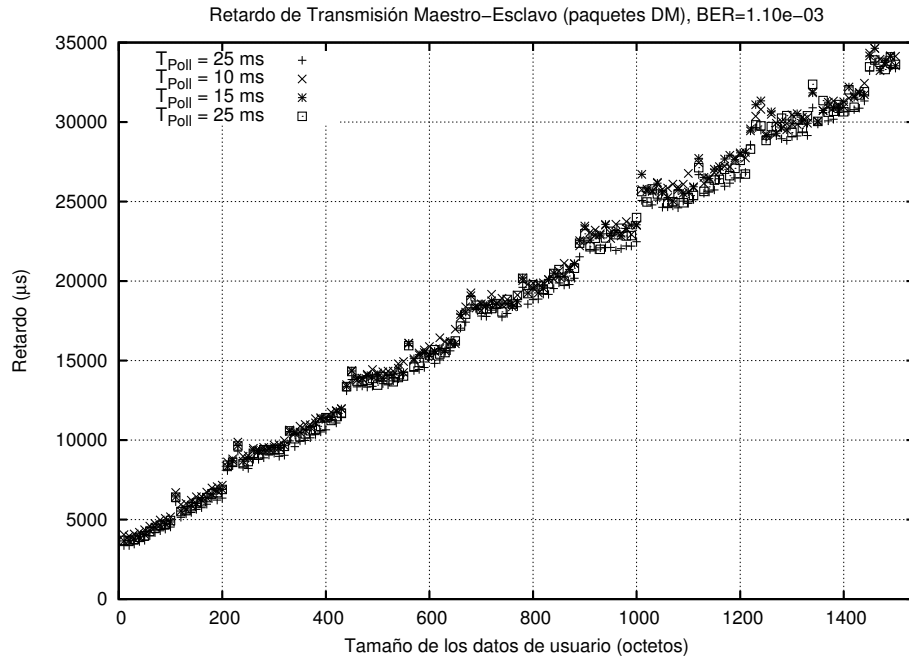
$$t_{off} = \begin{cases} 4 \cdot T_S & \text{Sentido maestro-esclavo} \\ \frac{T_{poll}}{2} + 3 \cdot T_S + \delta & \text{Sentido esclavo-maestro} \end{cases} \quad (6.2)$$

donde δ , que representa el *offset* que se añade para las longitudes de datos que exceden la máxima capacidad de un paquete de 5 *slots*, viene dado por la siguiente expresión:

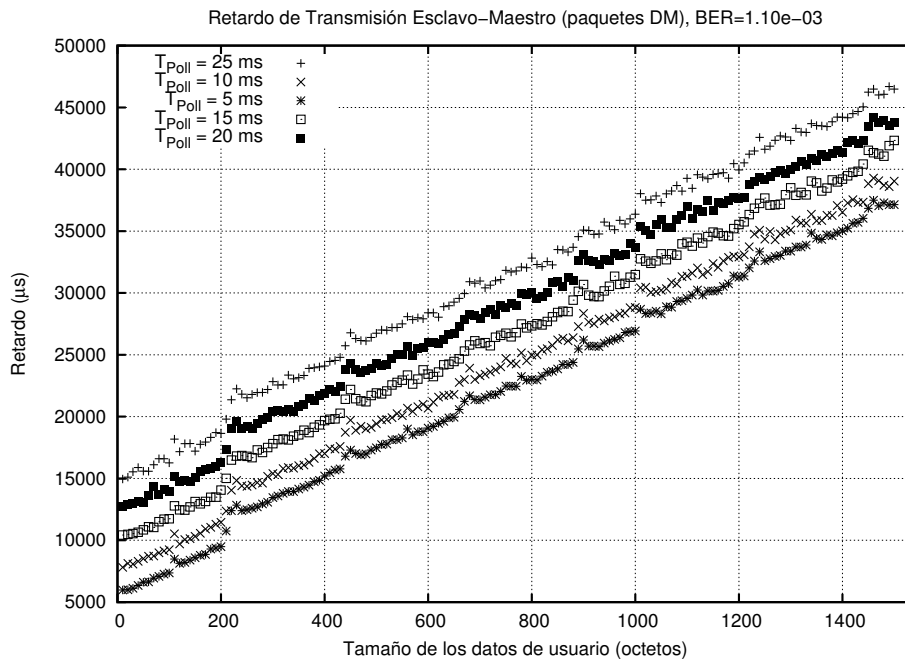
$$\delta = \begin{cases} 4 \cdot T_S & M(N) > L_5 \\ 0 & \text{en otro caso} \end{cases} \quad (6.3)$$

De este modo, se refleja el hecho de que si el esclavo envía paquetes de datos al ser direccionado por el maestro, el maestro continúa direccionándolo, pero no en el *slot* inmediatamente posterior a la recepción, sino incluso, en algunos casos, hasta cuatro *slots* más tarde.

La hipótesis que se formula como justificación de este fenómeno, que no tiene lugar en ausencia de retransmisiones, es la siguiente: cuando el maestro recibe un paquete



(a) Sentido de transmisión maestro-esclavo



(b) Sentido de transmisión esclavo-maestro

Figura 6.16.: Efecto de la configuración de la calidad de servicio (T_{poll}), en ambos sentidos de transmisión, para una atenuación de ruido de 6 dB.

de datos, la unidad de segmentación y ensamblado, SAR, decodifica la longitud de la cabecera del *payload* y la compara con la longitud de la trama de L2CAP, a fin de determinar si se ha recibido un fragmento único o el primero de una secuencia, para decidir si direcciona, o no, al esclavo en el siguiente *slot* maestro-esclavo. El maestro realizaría esta comprobación para todos los paquetes que transportaran un fragmento inicial de una trama *B-Frame* de L2CAP (LLID = 10, véase la tabla 2.4), poniéndose de manifiesto este comportamiento cuando la trama *B-Frame* de L2CAP es fragmentada para su transmisión, ya que el esclavo tiene que esperar la confirmación del maestro antes de proceder al envío del siguiente fragmento. A la vista de los resultados, esta confirmación parece demorarse hasta $4 \cdot T_{slot}$, tal vez por el tiempo requerido por el maestro para realizar esta comprobación [Moron08].

- El límite superior del intervalo dado para el término $t_{RTx}(x)$, que representa la separación entre retransmisiones, ha sido redefinido como se indica a continuación:

$$t_{RTx_{max}}(x) = T_{poll} + t_{RTx_{min}}(x) \quad (6.4)$$

La razón es contemplar que el temporizador que controla el mecanismo de *poll* ha de tener en cuenta la duración del paquete anteriormente transmitido. Es decir, el intervalo de *poll* ha de empezar a contarse a partir del siguiente *slot* de transmisión que corresponda:

- En el sentido maestro-esclavo: el siguiente *slot* a aquel en el que se recibe la confirmación negativa del esclavo (NAK), según se ilustra en la figura 6.18a.
- En el sentido esclavo-maestro: el *slot* inmediatamente posterior a la finalización de la recepción del paquete transmitido por el esclavo, tal y como se muestra en la figura 6.18b.

Nótese que de este modo sí se puede concebir, al menos teóricamente, un intervalo $T_{poll} = 1250 \mu s$, mínimo especificado en el estándar, ya que de otra manera sería imposible, al ser $1250 \mu s$ inferior a la duración de la transmisión de un paquete *multi-slot*.

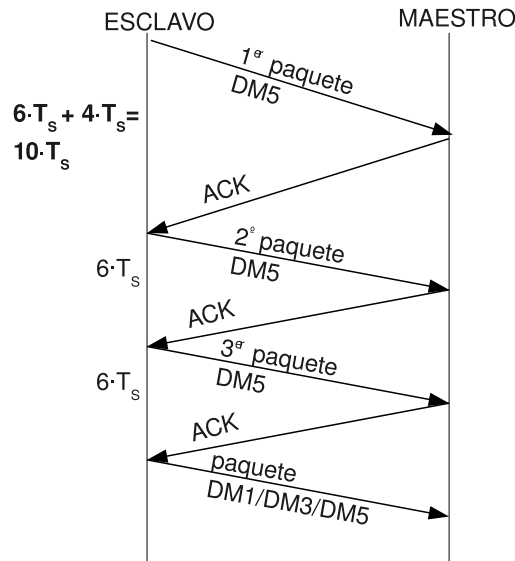


Figura 6.17.: Ejemplo que muestra la corrección introducida en el modelo para contemplar, en el sentido de transmisión esclavo-maestro, el salto incremental que se produce tras el llenado del primer paquete de 5 slots (fuente: [Moron08])

No obstante, se ha podido constatar que, en las pruebas realizadas sobre una *piconet* de dos dispositivos, la separación entre retransmisiones consecutivas no depende exactamente del intervalo T_{poll} configurado. Concretamente, $t_{RTx}(x)$ resulta tener un valor fijo independiente del sentido de transmisión y del intervalo de T_{poll} configurado: $16 \cdot T_s + t_{RTx_{min}}(x)$. Es decir, para todos los casos la distancia entre retransmisiones resulta ser el valor de $t_{RTx_{max}}(x)$ evaluado en $T_{poll} = 16 \cdot T_s$. Al respecto cabe destacar que, precisamente, 16 slots coincide con el mínimo intervalo T_{poll} que puede garantizar el maestro.

En las figuras 6.19a, 6.19b, 6.20a, 6.20b, 6.21a, 6.21b (en el sentido de transmisión maestro-esclavo), 6.22a, 6.22b, 6.23a, 6.23b, 6.24a, 6.24b, 6.25a y 6.25b (en el sentido de transmisión esclavo-maestro) se representa, para el rango de atenuación de ruido comprendido entre 4 y 8 dB, el retardo de transmisión medido y el teórico calculado mediante la ecuación 5.18, teniendo en cuenta que los términos t_{off} y $t_{RTx_{max}}(x)$ han sido redefinidos conforme a las ecuaciones 6.2 y 6.4. En concreto, se han representado las curvas teóricas correspondientes a la tasa BER medida (véase la tabla 6.1) y a las tasas BER

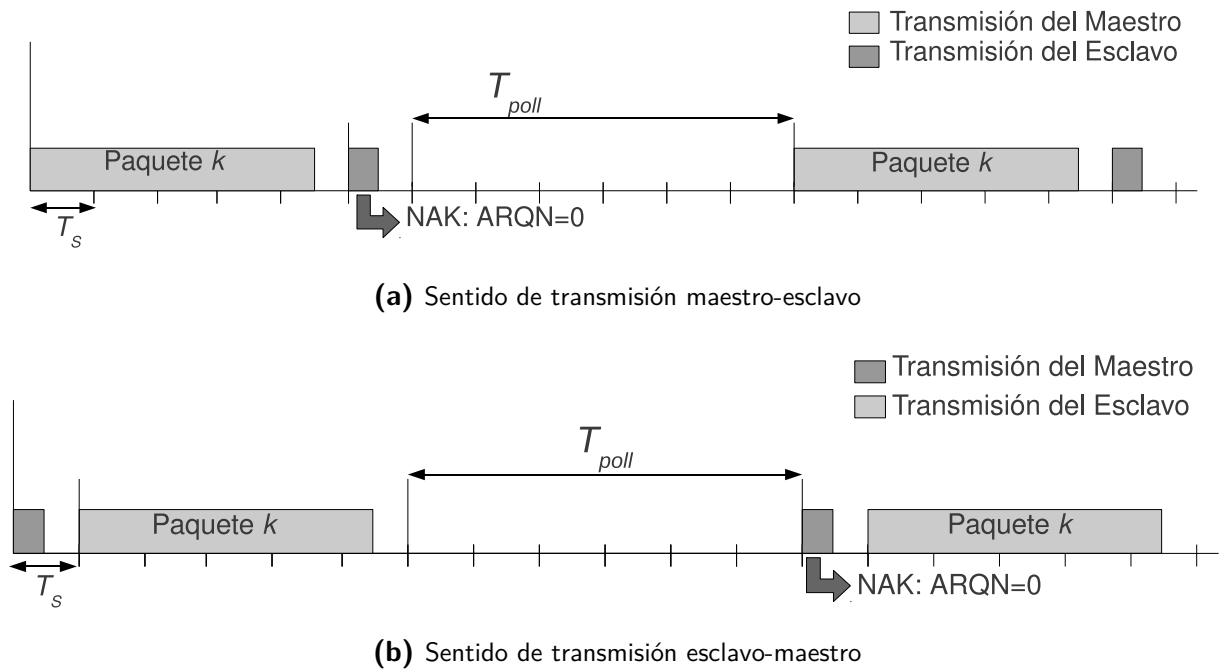


Figura 6.18.: Tiempo transcurrido desde que se inicia la transmisión de un paquete de 5 slots hasta que se realiza la retransmisión, particularizando $t_{RTx}(N)$ al valor mínimo, $t_{RTx_{min}}(N)$

obtenidas a partir de la ecuación 4.30 para los índices de modulación $h = 0,30$ y $h = 0,35$. Cabe destacar que:

- En todos los casos, el retardo obtenido a partir de la tasa BER medida por el procedimiento descrito en el apartado 6.2.1 ajusta con gran precisión el retardo medido empíricamente. El promedio de error relativo medio cometido en la estimación del retardo a partir de la tasa BER medida en el rango de relación SNR comprendido entre 10,95 y 14,95 dB, para cada uno de los sentidos de transmisión, se especifica en la tabla 6.4. En esta tabla puede observarse que el error medio está por debajo de un 6%.
- Conforme se incrementa la relación SNR el impacto del índice de modulación (h) en el modelo tiende a ser irrelevante. Nótese que, por ejemplo, en las figuras 6.21b (sentido maestro-esclavo) y 6.25b (sentido esclavo-maestro) las curvas teóricas son indistinguibles.

Tabla 6.4.: Error medio cometido en la estimación del retardo a partir de la tasa BER medida para valores de la relación SNR entre 10,95 y 14,95 dB, en ambos sentidos de transmisión.

SNR (dB)	BER	Maestro-Esclavo	Esclavo-Maestro
10,95	$5 \cdot 10^{-3}$	3,8 %	2,0 %
11,95	$2,3 \cdot 10^{-3}$	5,4 %	1,3 %
12,95	$1,1 \cdot 10^{-3}$	5,2 %	1,4 %
13,95	$4,8 \cdot 10^{-4}$	4,6 %	3,4 %
14,95	$1,3 \cdot 10^{-4}$	3,9 %	3,9 %

Además, en las gráficas correspondientes al sentido de transmisión esclavo-maestro puede constatar que, tal y como se ha indicado en epígrafes anteriores, el primer salto incremental que se produce en el retardo es notablemente mayor que los restantes. Es decir, se observa que el incremento en el retardo es mayor cuando el tamaño de la trama *B-Frame* de L2CAP, por exceder la capacidad de un paquete DM5, pasa de transportarse en un paquete de cinco *slots* a requerir un paquete adicional de un *slot*, que cuando se necesitan dos o más paquetes de cinco *slots*. No obstante, cabe destacar que el salto incremental del primer escalón, que ha sido computado en el término t_{off} , sumando un valor de 4 *slots* para longitudes de datos superiores a los 224 octetos que puede transportar como máximo un paquete DM5, se ajusta a la realidad en todos los casos.

Finalmente, en lo que respecta a las pruebas realizadas para el modo EDR queda decir que, pese a que se ha configurado un nivel de ruido relativamente bajo (atenuación de 8 dB), no ha sido posible comparar el retardo medido empíricamente con el obtenido para las tasas BER correspondientes a las modulaciones EDR. El motivo es que, aunque se fuerza el uso de paquetes EDR excluyendo la utilización de paquetes DM, el dispositivo, en respuesta al nivel de pérdidas detectado, pasa a emplear los paquetes DM de forma automática. Es conveniente aclarar que este hecho sólo se ha podido constatar a partir del resultado de la medida, ya que la curva del retardo presenta una pendiente de crecimiento del mismo orden que la que corresponde al uso de paquetes DM, tal y como se muestra en la gráfica de la figura 6.26, en la que se ha representado el retardo medido configurando el empleo de paquetes EDR y el teórico obtenido para paquetes DM, en las mismas condiciones descritas anteriormente.

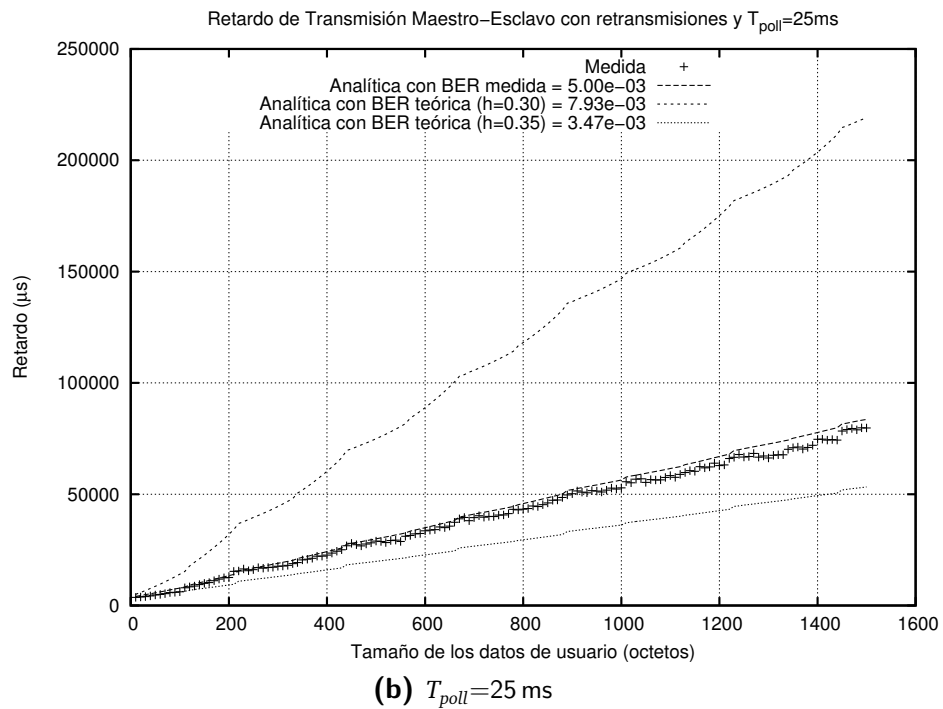
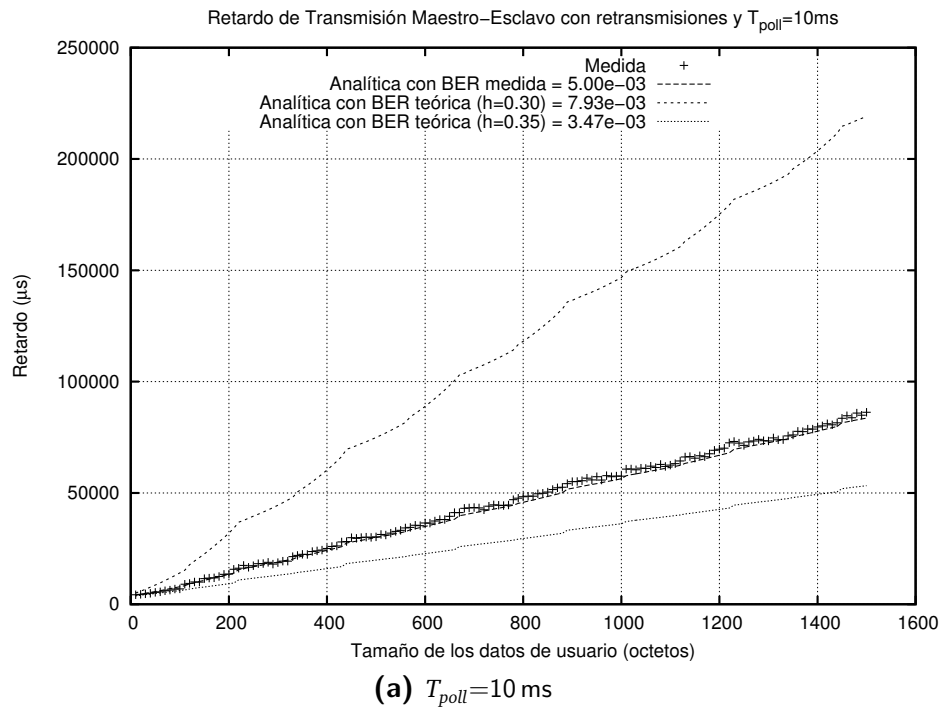
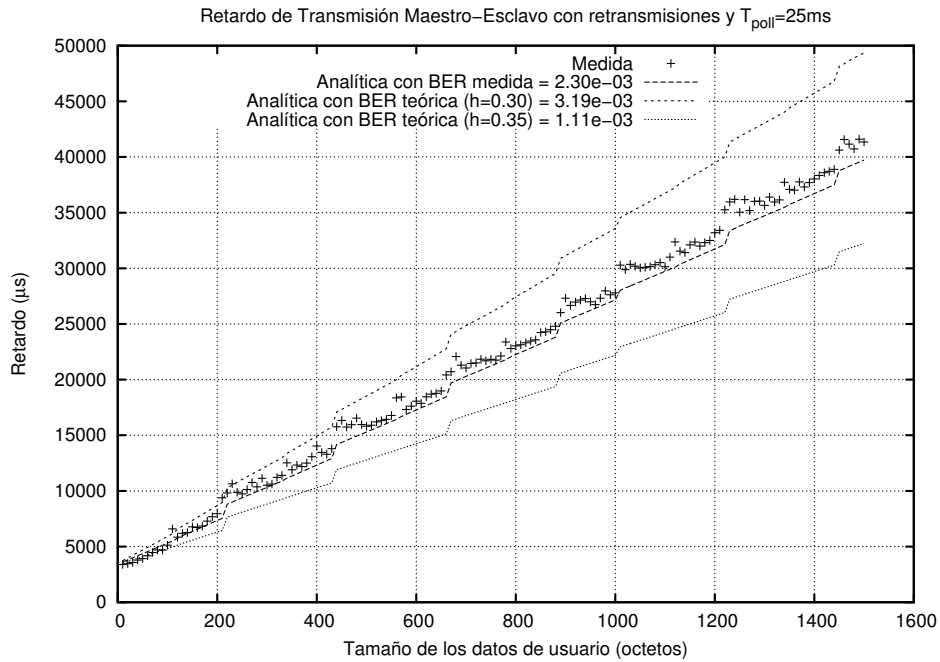
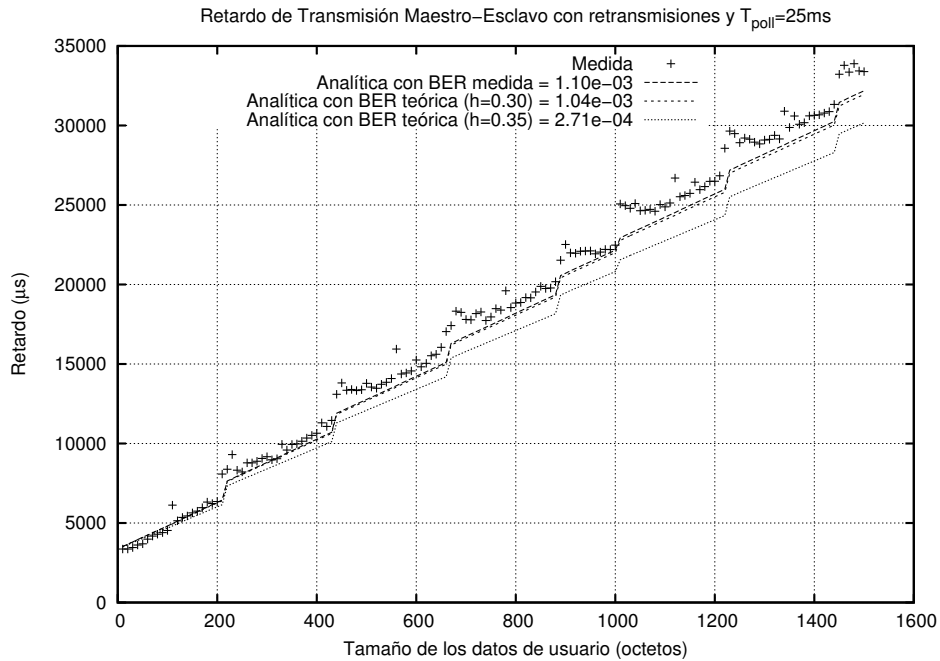


Figura 6.19.: Comparación del retardo medido y de las curvas teóricas calculadas a partir de la tasa BER medida y de las tasas BER teóricas obtenidas a partir de la relación SNR mediante la ecuación 4.30 para los índices de modulación $h = 0,30$ y $h = 0,35$, en el sentido maestro-esclavo, utilizando paquetes DM y para un nivel de atenuación de ruido de 4 dB ($\text{SNR} = 10,95\text{ dB}$).

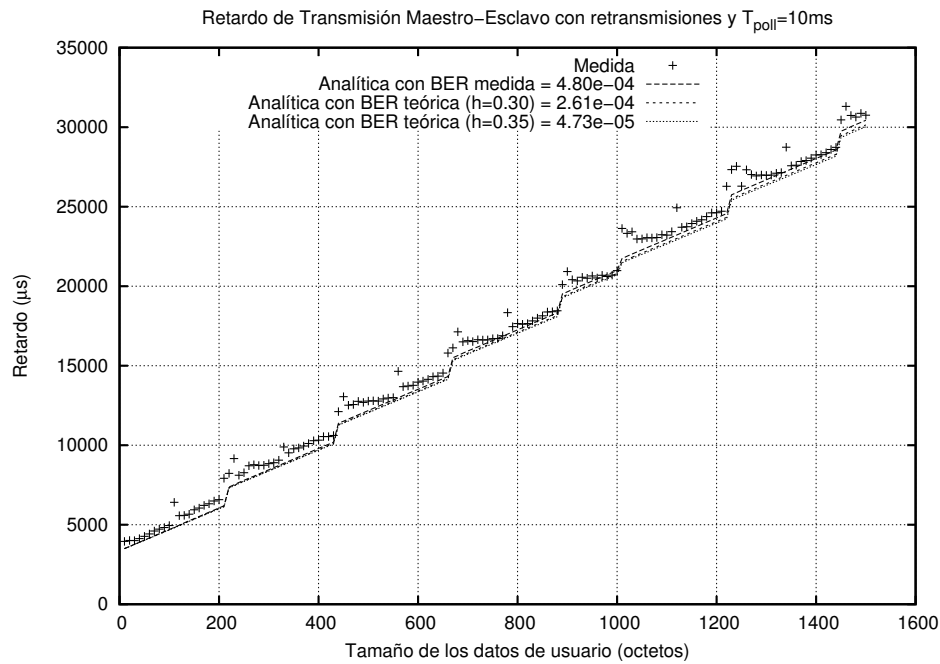


(a) Nivel de atenuación de ruido de 5 dB ($\text{SNR} = 11,95 \text{ dB}$) y $T_{\text{poll}}=25 \text{ ms}$

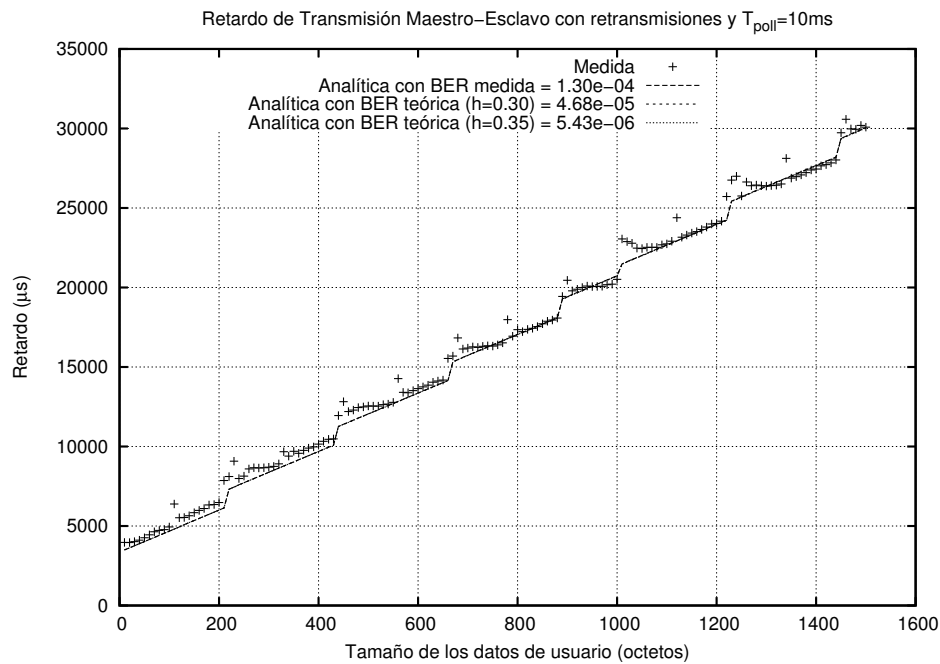


(b) Nivel de atenuación de ruido de 6 dB ($\text{SNR} = 12,95 \text{ dB}$) y $T_{\text{poll}}=25 \text{ ms}$

Figura 6.20.: Comparación del retardo medido y de las curvas teóricas calculadas a partir de la tasa BER medida y de las tasas BER teóricas obtenidas a partir de la relación SNR mediante la ecuación 4.30 para los índices de modulación $h = 0,30$ y $h = 0,35$, en el sentido maestro-esclavo, utilizando paquetes DM y para niveles de atenuación de ruido de 5 y 6 dB ($\text{SNR} = 11,95 \text{ dB}$ y $12,95 \text{ dB}$).

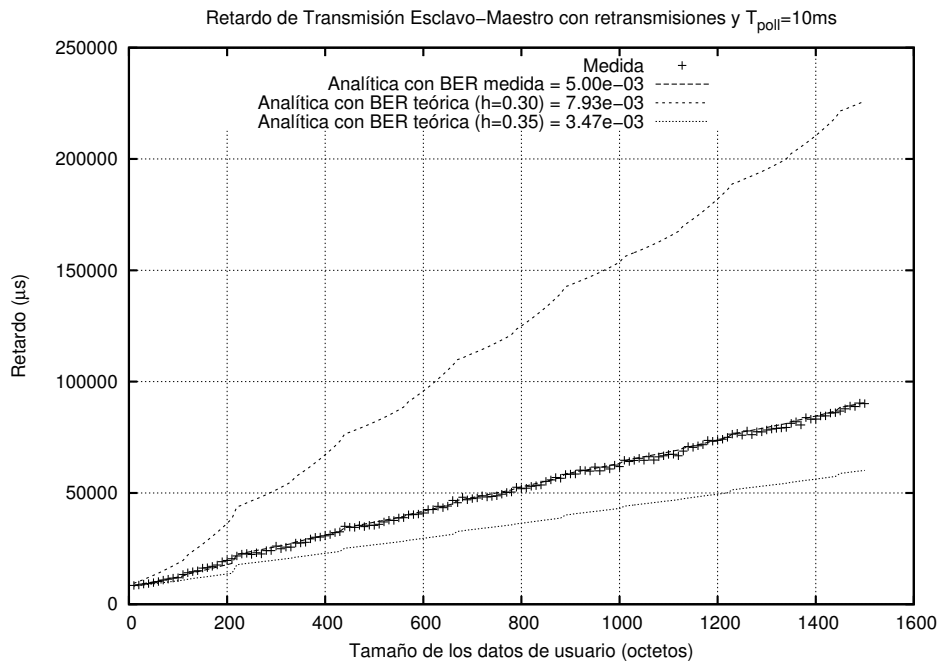


(a) Nivel de atenuación de ruido de 7 dB ($\text{SNR} = 13,95\text{ dB}$) y $T_{poll}=10\text{ ms}$

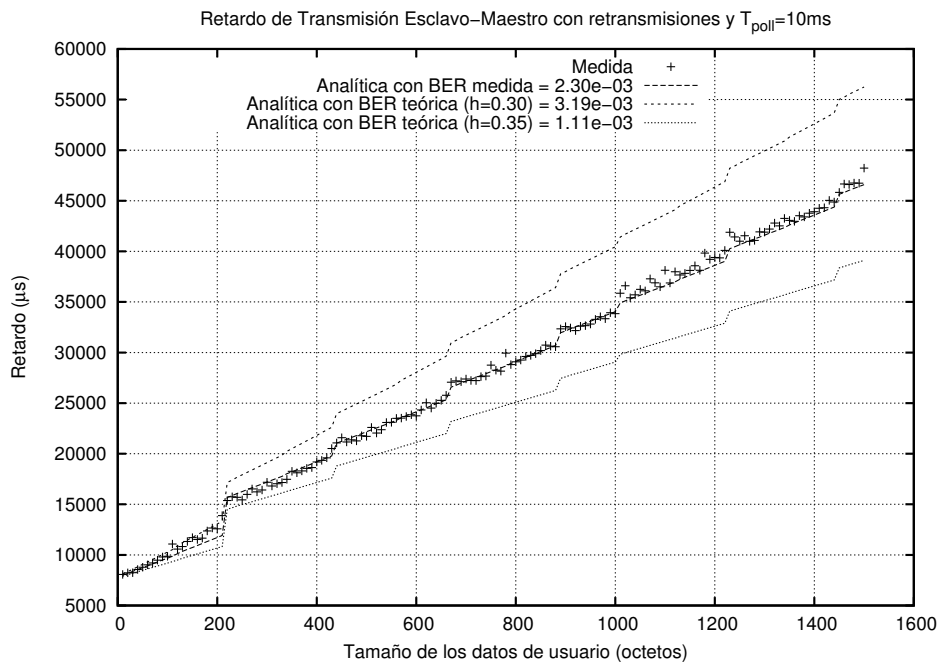


(b) Nivel de atenuación de ruido de 8 dB ($\text{SNR} = 14,95\text{ dB}$) y $T_{poll}=10\text{ ms}$

Figura 6.21.: Comparación del retardo medido y de las curvas teóricas calculadas a partir de la tasa BER medida y de las tasas BER teóricas obtenidas a partir de la relación SNR mediante la ecuación 4.30 para los índices de modulación $h = 0,30$ y $h = 0,35$, en el sentido maestro-esclavo, utilizando paquetes DM y para niveles de atenuación de ruido de 7 y 8 dB ($\text{SNR} = 13,95\text{ dB}$ y $14,95\text{ dB}$).

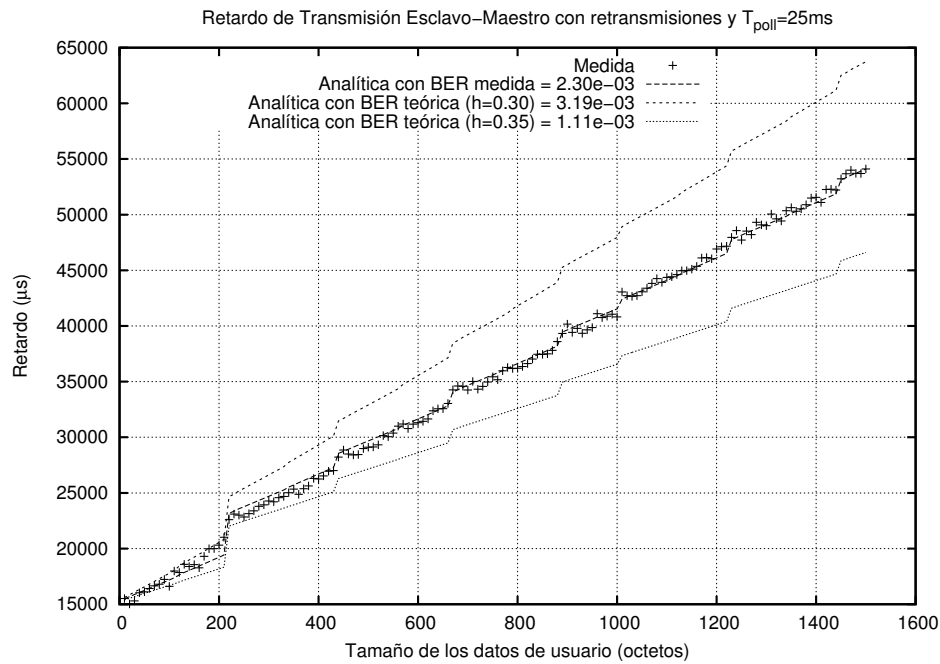


(a) Nivel de atenuación de ruido de 4 dB ($\text{SNR} = 10,95\text{ dB}$) y $T_{poll}=10\text{ ms}$

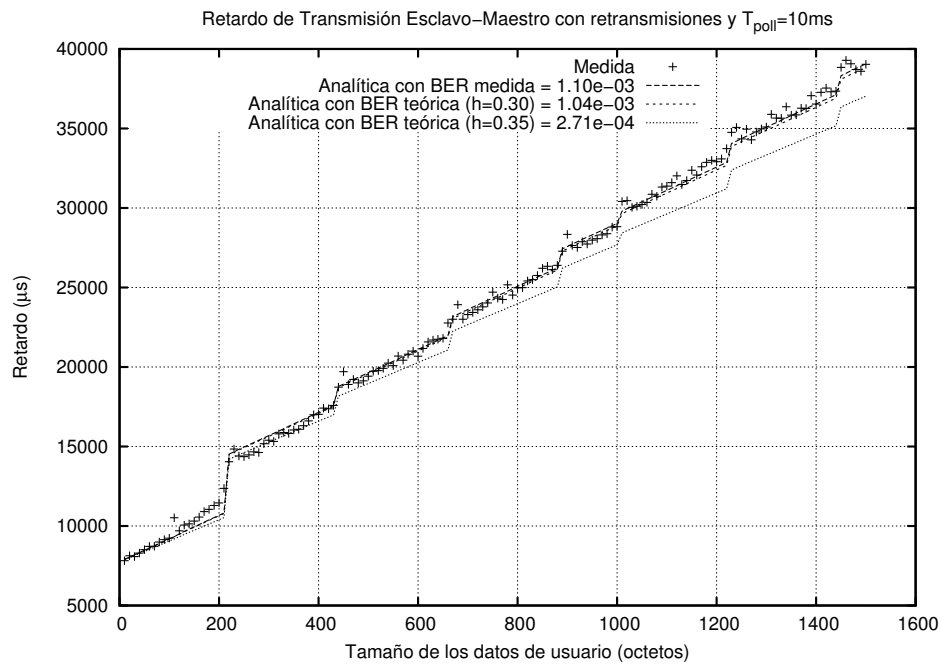


(b) Nivel de atenuación de ruido de 5 dB ($\text{SNR} = 11,95\text{ dB}$) y $T_{poll}=10\text{ ms}$

Figura 6.22.: Comparación del retardo medido y de las curvas teóricas calculadas a partir de la tasa BER medida y de las tasas BER teóricas obtenidas a partir de la relación SNR mediante la ecuación 4.30 para los índices de modulación $h = 0,30$ y $h = 0,35$, en el sentido esclavo-maestro, utilizando paquetes DM y para niveles de atenuación de ruido de 4 y 5 dB ($\text{SNR} = 10,95\text{ dB}$ y $11,95\text{ dB}$).



(a) Nivel de atenuación de ruido de 5 dB ($\text{SNR} = 11,95\text{ dB}$) y $T_{poll}=25\text{ ms}$



(b) Nivel de atenuación de ruido de 6 dB ($\text{SNR} = 12,95\text{ dB}$) y $T_{poll}=10\text{ ms}$

Figura 6.23.: Comparación del retardo medido y de las curvas teóricas calculadas a partir de la tasa BER medida y de las tasas BER teóricas obtenidas a partir de la relación SNR mediante la ecuación 4.30 para los índices de modulación $h = 0,30$ y $h = 0,35$, en el sentido esclavo-maestro, utilizando paquetes DM y para niveles de atenuación de ruido de 5 y 6 dB ($\text{SNR} = 11,95\text{ dB}$ y $12,95\text{ dB}$).

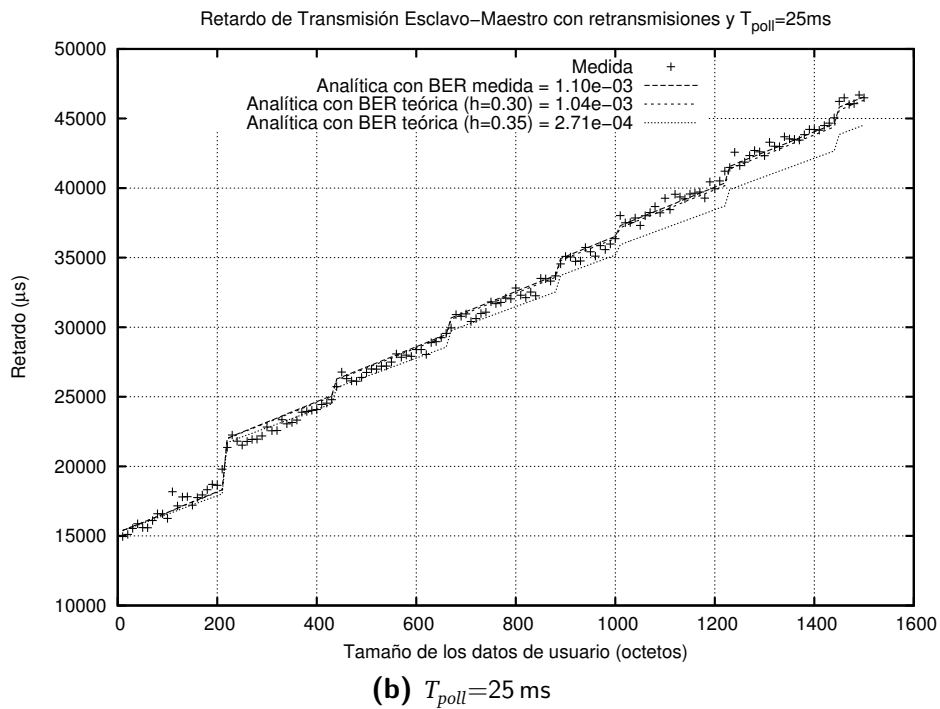
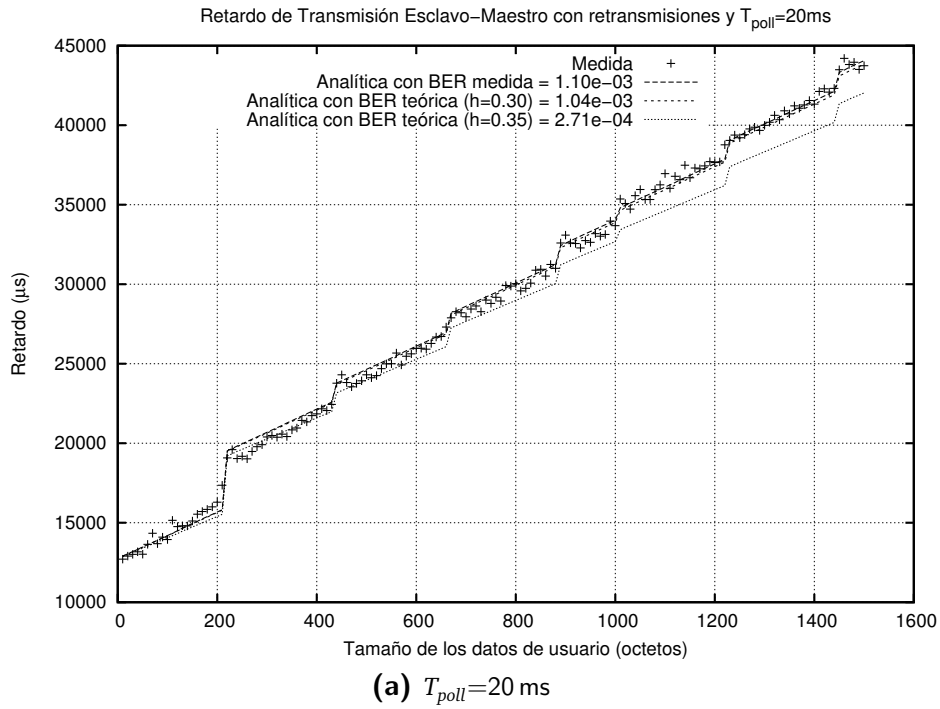


Figura 6.24.: Comparación del retardo medido y de las curvas teóricas calculadas a partir de la tasa BER medida y de las tasas BER teóricas obtenidas a partir de la relación SNR mediante la ecuación 4.30 para los índices de modulación $h = 0,30$ y $h = 0,35$, en el sentido esclavo-maestro, utilizando paquetes DM y para un nivel de atenuación de ruido de 6 dB ($\text{SNR} = 12,95\text{ dB}$).

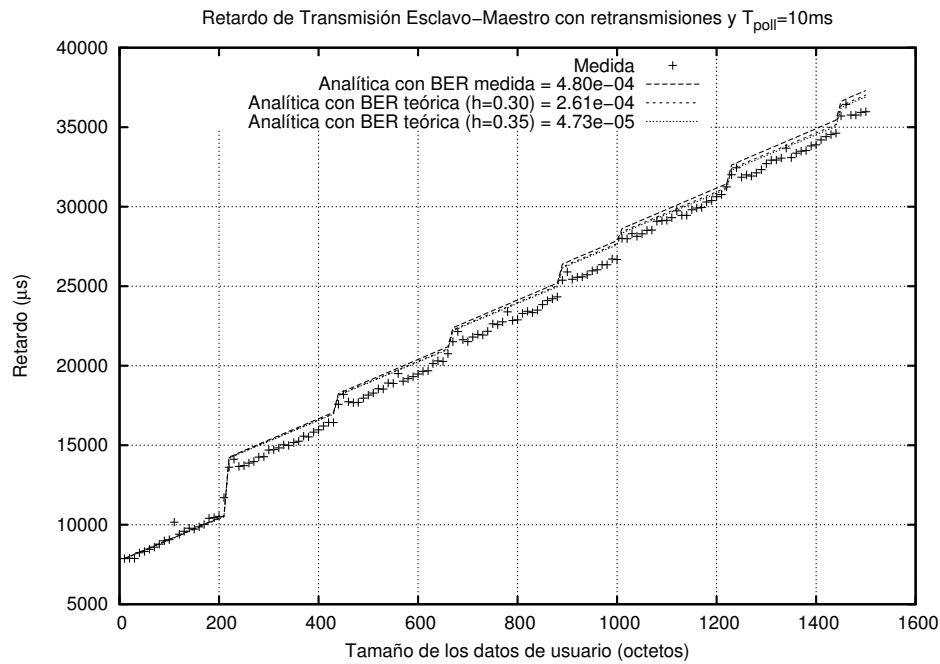
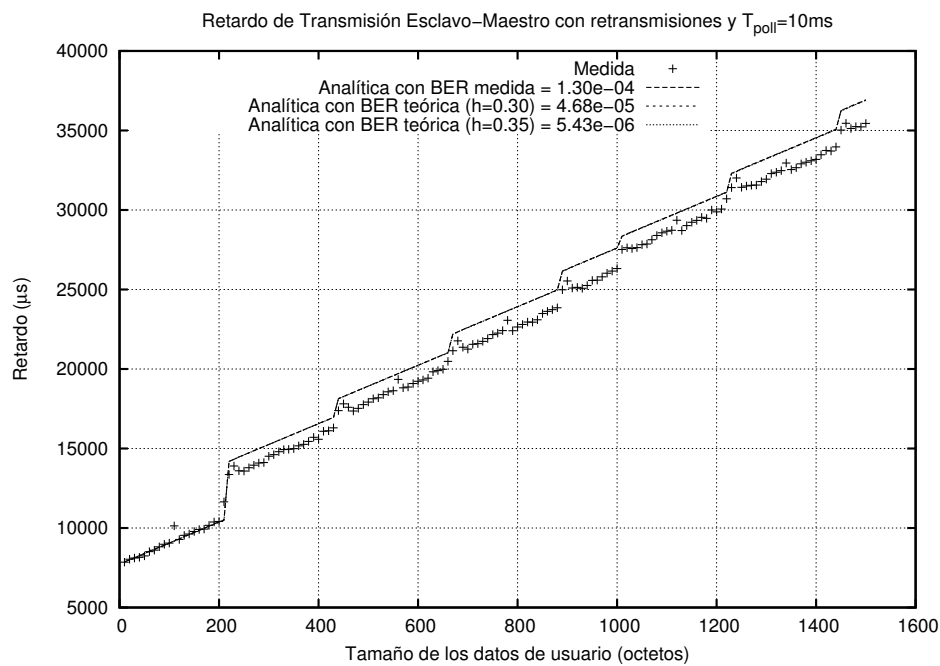
(a) Nivel de atenuación de ruido de 7 dB ($\text{SNR} = 13,95\text{ dB}$) y $T_{poll}=10\text{ ms}$ (b) Nivel de atenuación de ruido de 8 dB ($\text{SNR} = 14,95\text{ dB}$) y $T_{poll}=10\text{ ms}$

Figura 6.25.: Comparación del retardo medido y de las curvas teóricas calculadas a partir de la tasa BER medida y de las tasas BER teóricas obtenidas a partir de la relación SNR mediante la ecuación 4.30 para los índices de modulación $h = 0,30$ y $h = 0,35$, en el sentido esclavo-maestro, utilizando paquetes DM y para niveles de atenuación de ruido de 7 y 8 dB ($\text{SNR} = 13,95\text{ dB}$ y $14,95\text{ dB}$).

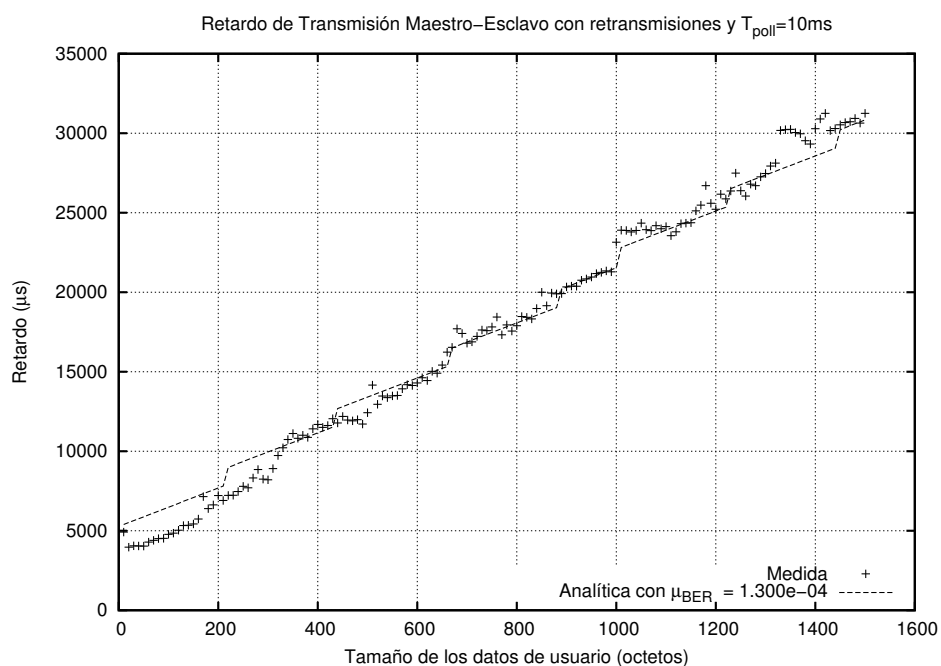


Figura 6.26.: Comparación del retardo medido en el modo EDR y de la curva teórica calculada para paquetes DM a partir de la tasa BER medida, en el sentido maestro-esclavo, para un nivel de atenuación de ruido de 8 dB y $T_{poll}=10\text{ ms}$.

CAPÍTULO 7

Conclusiones y líneas futuras

Las conclusiones extraídas del desarrollo de esta tesis se recopilan en la sección 7.1. Después, en la sección 7.2, se incluyen las líneas en las que se propone continuar investigando. Finalmente, en la sección 7.3, se relacionan las publicaciones derivadas de la realización de esta tesis.

7.1. Conclusiones

El objetivo principal de la presente tesis ha sido modelar el retardo de transmisión para el perfil de puerto serie SPP (*Serial Port Profile*) en Bluetooth 2.0 + EDR, tanto para el modo básico, compatible con Bluetooth 1.1, como para el modo EDR (*Enhanced Data Rate*). El estudio desarrollado ha consistido en:

- Formular expresiones matemáticas para calcular el retardo de transmisión, en los sentidos maestro-esclavo y esclavo-maestro.

La estimación teórica del retardo de transmisión se ha llevado a cabo, en primer lugar, asumiendo condiciones ideales, es decir, considerando que no se producen retransmisiones a nivel Baseband (BB), para a continuación extender el modelo considerando el efecto de las posibles pérdidas en las transmisiones.

- Validar empíricamente el modelo desarrollado utilizando la pila de protocolos BlueZ y dispositivos reales de distintos fabricantes de gran implantación en productos Bluetooth comerciales: CSR (*Cambridge Silicon Radio*), ISS (*Integrated System Solution Corp.*) y *Broadcom Corporation*.

En el caso de los dispositivos CSR se ha realizado un análisis exhaustivo, comparando el retardo previsto teóricamente con el obtenido empíricamente e introduciendo las correcciones necesarias para contemplar dos fenómenos que distorsionan el retardo medido, enmascarando la validez del modelo formulado: el efecto de la capa de transporte y de la fragmentación según el tamaño de los *buffers* del dispositivo.

Sin embargo, para los dispositivos ISS y *Broadcom Corporation*, el análisis no ha podido ser tan detallado, debido a que no se disponen de las mismas herramientas que en el caso de CSR y a que se ha comprobado que los propios dispositivos no funcionan correctamente en todos los casos.

Seguidamente, en el apartado 7.1.1, tras resumir las bases del modelo teórico definido para Bluetooth 2.0 + EDR sin retransmisiones, se exponen las conclusiones sobre la validación del modelo llevada a cabo con dispositivos CSR (apartado 7.1.1.1), ISS (apartado 7.1.1.2) y *Broadcom Corporation* (apartado 7.1.1.3). Después, en el apartado 7.1.2 se presentan los fundamentos teóricos de la extensión del modelo para contemplar el efecto de las retransmisiones y se sintetizan tanto las conclusiones derivadas del análisis teórico como de la validación empírica realizada con dispositivos CSR.

7.1.1. Conclusiones sobre el modelo teórico sin retransmisiones

Dado que los protocolos de los niveles L2CAP (*Logical Link Control and Adaptation Protocol*) y RFCOMM (*Serial Cable Emulation Protocol*, basado en la norma ETSI TS 07.10) no se ven modificados sustancialmente en la especificación Bluetooth 2.0 + EDR, en relación con el modelo presentado en [Moron08], el cálculo del retardo de transmisión para el perfil SPP, requiere reformular únicamente las componentes $t_{ACK}(x)$ y $t_{TX}(x)$, términos que tienen en cuenta la fragmentación realizada a nivel BB de un mensaje del nivel superior. Para ello, es necesario incorporar en el modelo la existencia de los paquetes específicos de los modos

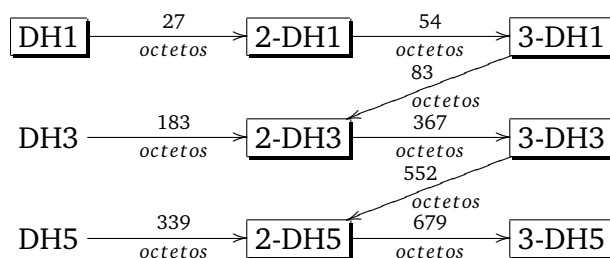


Figura 7.1.: Esquema de utilización de paquetes BB según se incrementa el tamaño de los datos de usuario

EDR. Concretamente, el esquema de selección de paquetes DH EDR que sustenta el modelo desarrollado sigue la secuencia mostrada en el esquema de la figura 7.1.

Así, la política propuesta postula que, conforme se incrementa el tamaño de los datos de usuario, se aumenta la velocidad de modulación antes de incrementar el número de *slots* ocupados, con la salvedad de los paquetes DH3 y DH5, que parecen no emplearse.

Con este esquema de elección de tipos de paquetes, el principal efecto que introduce el modo EDR al aumentar el tamaño de los datos a transmitir es la presencia en la curva del retardo de incrementos y decrementos bruscos, asociados a los cambios de la velocidad de modulación.

■ Decrementos (2 Mbps $\rightarrow$ 3 Mbps):

- Cuando se excede la capacidad de un paquete 2-DH3, 367 octetos, y pasa a utilizarse un paquete 3-DH3 (2-DH3 $\rightarrow$ 3-DH3).
- Al superarse los 679 octetos que, como máximo, pueden transportarse en un paquete 2-DH5 y, por tanto, se conmuta a un paquete 3-DH5 (2-DH5 $\rightarrow$ 3-DH5).

■ Incrementos bruscos (3 Mbps $\rightarrow$ 2 Mbps):

- Tras completarse un paquete 3-DH1, con una capacidad de 83 octetos, y conmutar a un paquete 2-DH3 (3-DH1 $\rightarrow$ 2-DH3).

- Cuando la longitud de los datos es superior al número de octetos que pueden transportarse en un paquete 3-DH3, 552 octetos, valor para el que se incrementa el número de *slots*, conmutando a un paquete de menor velocidad: 2-DH5 (3-DH3 → 2-DH5).
- Al excederse la máxima capacidad de un paquete 3-DH5, 1021 octetos, límite que viene a coincidir, aproximadamente, con los 1017 octetos de una trama L2CAP encapsulando una trama RFCOMM completa con 1008 octetos de usuario, por encima de los cuales se pasa a requerir más de una trama RFCOMM para su transmisión.

En lo que respecta a los paquetes DM no se produce cambio alguno respecto a la versión Bluetooth 1.1. Es decir, el retardo tiene un comportamiento creciente de forma que cada vez que se produce el llenado de un paquete DM5 tiene lugar un salto ascendente, de 6 *slots*, debido a que para mandar un paquete que transporte un fragmento no inicial hay que esperar la confirmación del anterior.

A continuación se presentan las conclusiones derivadas de la validación empírica del modelo desarrollado.

7.1.1.1. Dispositivos CSR

En este apartado se recopilan las conclusiones extraídas sobre la evaluación del modelo realizada con dispositivos del fabricante CSR de la serie *BlueCore 4*. En lo que concierne al modelo teórico, el comportamiento de estos dispositivos se caracteriza por:

1. La asimetría del funcionamiento del mecanismo de *poll*, regulada por el periodo T_{poll} :
 - Sentido esclavo-maestro: El esclavo se ve obligado a esperar que el maestro de la *piconet* lo direcciona mediante el mecanismo de *poll*, de promedio cada $T_{poll}/2$ en el modo EDR, tiempo al que se suman dos *slots* en el modo básico, incrementándose, por tanto, el retardo de transmisión de datos en esta cantidad.

- Sentido maestro-esclavo: No existe dependencia del intervalo T_{poll} configurado, es decir, el maestro, que gestiona el mecanismo de *poll*, no espera un tiempo variable de forma que el retardo adicional es fijo, siendo, en promedio, de dos *slots* para el modo EDR, y de tres *slots* para el modo básico.

Adicionalmente, se ha identificado la influencia de dos efectos en las medidas realizadas que, en una primera aproximación, enmascaran la validez del modelo desarrollado:

2. El retardo asociado a la capa de transporte USB (*Universal Serial Bus*), que resulta ser independiente del sentido de transmisión, y está presente tanto en dispositivos Bluetooth 2.0+ EDR con versión de *firmware* 0x7a6 y 0xc5c, como en dispositivos Bluetooth 2.1 + EDR.
3. El retardo introducido por la fragmentación realizada de acuerdo con el valor del parámetro ACL MTU, efecto que tiene lugar tanto en el modo básico como en el modo EDR, para el sentido de transmisión maestro-esclavo y en dispositivos Bluetooth 2.0 + EDR, a partir de la versión de *firmware* 0xc5c.

Para evaluar el efecto sobre el retardo de la transferencia USB, ha sido necesario caracterizar la capa de transporte. Como resultado de este análisis:

- Aunque cabría esperar un retardo linealmente creciente con el número de octetos, se ha detectado que existe un límite, 512 octetos, a partir del que se produce un incremento abrupto del retardo (de 1,20 ms), diferenciándose dos tramos con pendientes de crecimiento distintas.
- Se ha obtenido empíricamente una ecuación matemática que aproxima el retardo registrado que introduce la capa de transporte.

En relación, con ambos efectos se han extraído las conclusiones siguientes para ambos modos de funcionamiento:

- La fragmentación en bloques de 512 octetos, realizada para la transferencia USB, no influye en el esquema de transmisión, ya que afecta únicamente a la comunicación entre el *Host* y el *Host Controller*.

- La fragmentación que se lleva a cabo atendiendo al valor del parámetro ACL MTU (M_{ACL}) condiciona:
 - La transferencia a través de la capa de transporte.
 - Los paquetes BB utilizados en la transmisión Bluetooth, de forma que si la longitud de los datos a transmitir es superior al valor del parámetro ACL MTU y ésta, a su vez, es inferior a la máxima capacidad de un paquete de 5 *slots*, no se va a conseguir el máximo rendimiento porque se va a transmitir un primer paquete con M_{ACL} octetos (ni siquiera completo) y, a continuación, un segundo paquete (o incluso más) con el número de octetos en los que la longitud de los datos de usuario excede el límite impuesto por el parámetro ACL MTU.
- La principal consecuencia de la combinación de los efectos —fragmentación al valor del parámetro ACL MTU y transferencia por USB— es una pérdida de rendimiento. Concretamente, para maximizar el rendimiento, minimizando el retardo de transmisión, para cada uno de los modos se han de cumplir las siguientes condiciones:
 - Modo básico: El parámetro ACL MTU se configura a un valor igual o superior a la máxima capacidad de un paquete de 5 *slots*.
 - Modo EDR: El valor del parámetro ACL MTU coincide con el límite impuesto por la capa de transporte, 512 octetos.

Por último, en lo que concierne al análisis del comportamiento de la capa de transporte HCTL USB, cabe resaltar que su aplicación no está limitada al estudio abordado en esta tesis con módulos Bluetooth externos. De hecho, incluso dispositivos con tecnología Bluetooth 2.0 + EDR, como teléfonos móviles, incluyen una interfaz USB interna como capa HCTL para la comunicación con el módulo (*Host Controller*), debido a que, a diferencia de las restantes interfaces, proporciona un ancho de banda superior a la máxima tasa de transferencia del modo EDR.

7.1.1.2. Dispositivos ISS

En lo que concierne al efecto de la configuración de la calidad, de forma similar a lo que sucede con los dispositivos CSR, se ha detectado una asimetría en el funcionamiento del mecanismo de *poll*, de forma que:

- En el sentido maestro-esclavo: El resultado es el mismo en todos los casos, independientemente de que se configure o no, la calidad de servicio, y del valor del parámetro del periodo T_{poll} configurado.
- En el sentido esclavo-maestro: Existe dependencia del parámetro T_{poll} fijado, aunque se alcanzan retardos de transmisión excesivamente elevados, del orden de diez veces superior a los obtenidos en el sentido maestro-esclavo.

En cuanto al esquema de transmisión y recepción EDR utilizado por los dispositivos ISS cabe destacar que:

1. Una vez que se excede la capacidad de un paquete 2-DH3 (367 octetos), no se produce un decremento brusco del retardo como resultado del cambio de la modulación $\pi/4$ DQPSK a 8 DPSK.
2. No tiene lugar decremento alguno en el caso en el que, teóricamente, se aumenta la tasa de 2 Mbps a 3 Mbps, cuando se pasa de emplear un paquete 2-DH5 a un paquete 3-DH5.
3. Los incrementos abruptos del retardo se producen en torno a las longitudes a las que se llenan los paquetes de máxima velocidad para un determinado número de *slots*: 83 octetos (3-DH1), 552 octetos (3-DH3) y 1021 octetos (3-DH5).

La razón que justifica estos resultados no es la misma en los tres casos:

- Casos (1) y (2): El módulo radio no entrega los datos al *Host* Bluetooth hasta que no se completa el número de *slots* ocupados por el paquete en el que se reciben los datos. Sin embargo, el retardo no es constante, como cabría esperar, sino que presenta un comportamiento creciente debido a que la medida realizada, al igual que con los dispositivos CSR, incluye el efecto de la capa de transporte.

- Caso (3): El agotamiento de la capacidad de un paquete 3-DH5 coincide con el tamaño de los datos de usuario cuyo envío pasa a requerir dos tramas RFCOMM, punto a partir del cual, el comportamiento real sólo se diferencia del previsto teóricamente en cierto retardo constante u *offset*.

Dado que el comportamiento de la capa de transporte no ha podido modelarse como en el caso de los dispositivos CSR, se ha inferido restando al retardo medido empíricamente el obtenido teóricamente, suponiendo que el módulo radio no entrega los datos al *Host* Bluetooth hasta que no se completa el número de *slots* ocupados por el paquete en el que se reciben los datos. A partir de esta suposición, se concluye que la transferencia USB:

- Introduce un retardo que crece linealmente con el tamaño de los datos, para longitudes inferiores a 1008 octetos
- Añade un retardo constante para longitudes superiores a 1008 octetos.
- No presenta discontinuidad alguna a la longitud de 512 octetos, a diferencia de lo que sucede con los dispositivos del fabricante CSR.

Por último, en lo que respecta a los resultados medidos al configurar los dispositivos ISS para utilizar exclusivamente paquetes DM, se concluye que no son operativos, debido a que una vez que los datos de usuario superan los 1250 octetos el retardo degenera de forma significativa, ya que, en un escenario sin pérdidas, se alcanzan retardos superiores a 100 ms para el envío de más de 1250 octetos de usuario.

7.1.1.3. Dispositivos Broadcom

El retardo medido en el sentido de transmisión maestro-esclavo, sin configurar calidad de servicio, y empleando paquetes DH del modo básico y EDR, no sólo no responde al comportamiento esperado sino que se caracteriza por una gran inestabilidad y dispersión, siendo incluso peor en el sentido esclavo-maestro.

Por contra, cuando se configuran los dispositivos para utilizar exclusivamente paquetes DM los retardos medidos, a diferencia de lo que sucede con dispositivos ISS, responden al modelo teórico, aunque no al básico, sí al modificado considerando que el retardo calculado

por la componente $t_{TX}(x)$ no depende exactamente del número de bits a transmitir sino del número de *slots* ocupados por el paquete que se utiliza.

7.1.2. Conclusiones sobre el modelo teórico con retransmisiones

El hecho de que Bluetooth opere en la banda ISM (*Industrial, Scientific and Medical*), compartida con otras tecnologías inalámbricas, convierte a los escenarios sin pérdidas en poco realistas. Por este motivo, se ha considerado imprescindible completar el modelo sin pérdidas para contemplar el efecto de las retransmisiones en el cálculo del retardo asociado al perfil SPP. La extensión se ha llevado a cabo para el modo de operación básico del protocolo L2CAP (modo por defecto), en el que no se realizan retransmisiones a nivel L2CAP y que es el que generalmente se utiliza en las aplicaciones de Bluetooth, debido a que garantiza la interoperatividad con versiones anteriores a Bluetooth 2.0 + EDR. Así pues, el objetivo perseguido ha sido incorporar el efecto de las retransmisiones a nivel BB en el cálculo del retardo.

A tal fin, la extensión del modelo ha supuesto la redefinición de las ecuaciones correspondientes a las dos componentes, $t_{ACK}(x)$ y $t_{TX}(x)$. Dichas componentes son calculadas a partir del número medio de retransmisiones, obtenido en función de la tasa de paquetes erróneos (PER) independientemente de que se trate del modo básico o del modo EDR. Por su parte, la tasa PER se determina en función de:

- El número de bits del campo de datos en el paquete Bluetooth.
- La tasa de bits erróneos, BER, que, para las mismas condiciones del canal de transmisión, modeladas mediante cierta relación señal/ruido, SNR (*Signal Noise Ratio*), no es igual si emplea la modulación básica o alguna de las modulaciones EDR: $\pi/4$ DQPSK o 8 DPSK.

En cualquiera de los casos, la obtención de la tasa BER, para cada una de las modulaciones, a partir de la SNR, se ha llevado a cabo asumiendo que el canal de transmisión Bluetooth se puede modelar como un canal de ruido blanco gaussiano (AWGN), es decir, considerando que el efecto del ruido es equivalente a la adición de ruido gaussiano de potencia constante en la banda considerada.

En definitiva, el número de retransmisiones, $\overline{N_{Tx}}$, se determina a partir de la relación SNR por bit (γ_b) y, en función de la modulación.

A continuación, en los siguientes apartados se presentan las conclusiones extraídas a partir del análisis de modelo teórico (apartado 7.1.2.1) y de su validación (7.1.2.2).

7.1.2.1. Conclusiones derivadas del modelo teórico

Del análisis teórico llevado a cabo se infiere que, cuando se utiliza la modulación 8 DPSK, el retardo es muy elevado al tener un comportamiento mucho peor que el resto de las modulaciones para una determinada relación SNR. A esta conclusión conducen los siguientes hechos:

- Para una probabilidad de error de bit P_b de 10^{-3} , la modulación 8 DPSK precisa una SNR por bit (γ_b) en torno a 17,5 dB, mientras que las modulaciones GFSK y $\pi/4$ DQPSK tan sólo requieren para la SNR un valor de γ_b sobre 12,1 dB.
- Suponiendo como valor límite de la probabilidad de error de bit P_b de 10^{-2} , a partir del cual se compromete seriamente el propio mantenimiento del enlace, se observa que en el caso de 8 DPSK la SNR debe ser superior a 15,2 dB, en contraste con los 10 dB necesarios para GFSK y $\pi/4$ DQPSK.

Adicionalmente, del análisis realizado se han inferido criterios de decisión sobre la utilización de las diferentes modulaciones y tipos de paquetes, considerando la política de selección reflejada en la figura 7.1, para longitudes inferiores a la máxima capacidad de un paquete DM5, 224 octetos, y los siguientes intervalos de la relación SNR:

- $\gamma_b \leq 8,5$ dB: No es viable mantener el enlace, independientemente del tipo de paquete empleado.
- $8,5 < \gamma_b \leq 10$ dB: Sólo es factible emplear paquetes DM, ya que el uso de los paquetes DH del modo básico y EDR conlleva retardos intolerables o, incluso, la pérdida del enlace.

- $10 < \gamma_b \leq 13,8$ dB: Es aceptable la utilización de paquetes DH y 2-DH, aunque entre ambos, 2-DH es más eficiente a partir de 13 dB. Sin embargo, sigue siendo óptima la transmisión de información en paquetes DM.
- $13,8 < \gamma_b \leq 19$ dB: Es óptima la utilización de paquetes 2-DH y, en su defecto, es admisible optar por paquetes DH del modo básico. Únicamente, a partir de 16 dB, se puede considerar el uso de paquetes 3-DH.
- $\gamma_b > 19$ dB: A partir de este límite puede empezar a considerarse óptima la transmisión de información empleando paquetes 3-DH.

A modo de conclusión, cabe destacar la utilidad de los paquetes DM, con codificación FEC $2/3$, ya que permiten operar, incluso, con tasas BER superiores al 1 %. Así, mientras que para los paquetes DM, retardos del orden de 1 s corresponden a una relación SNR de 9,5 dB y a una tasa BER del 1,31 %, en el caso de emplear paquetes DH, este orden de magnitud en el retardo se alcanza con una relación SNR de, al menos, 12 dB, que tiene asociada una tasa BER del 0,11 %.

7.1.2.2. Conclusiones sobre la validación empírica del modelo con retransmisiones

Para la validación empírica del retardo teórico calculado para el perfil SPP considerando el efecto de las retransmisiones se ha diseñado un escenario que permite reproducir empíricamente el modelo de canal de canal gaussiano propuesto teóricamente. Para ello, se ha establecido un camino conducido entre los transceptores de los dos dispositivos, conectándolos, mediante cable coaxial, a un generador de ruido blanco y a un atenuador.

Los dispositivos comerciales empleados como unidades Bluetooth en todas las pruebas realizadas para la validación empírica del cálculo del retardo sin retransmisiones han sido reemplazados por placas de desarrollo, del fabricante BlueGiga (*Kit de evaluación EKWT11-E*), que incorporan el mismo *chipset* de fabricante CSR, *BlueCore 4*.

Con este marco de prueba, previamente a la validación empírica del retardo, se ha realizado también la validación, también empírica, de las tasas BER calculadas teóricamente. Las conclusiones extraídas en cada caso se exponen en los siguientes apartados.

Conclusiones sobre la validación empírica de las tasas BER calculadas teóricamente

Con los dispositivos empleados, a diferencia de lo que sucede con dispositivos Bluetooth 1.1, no ha sido posible utilizar el comando *HCI_Read_Link_Quality* para obtener el parámetro *Link Quality*, directamente relacionado con la tasa BER. Puesto que con este método no ha sido factible obtener valores válidos para la tasa BER, se ha optado por utilizar, desde la aplicación *BlueTest* (proporcionada por el fabricante), las capacidades de prueba de la capa RF implementadas en los módulos Bluetooth para medir la tasa BER correspondiente a diferentes niveles de atenuación de ruido.

Por otro lado, se ha utilizado el analizador de espectro de Agilent Technologies modelo N9010A [Agilent Tech.c] para medir la potencias de señal y ruido reales de forma que, una vez determinada la relación SNR y, en función de esta, se compara la tasa BER obtenida a partir de las ecuaciones teóricas con la medida.

De la comparación de las tasas BER medidas con las calculadas de forma teórica, obtenidas a partir de la relación SNR obtenida experimentalmente, se concluye que:

■ Modulación GFSK:

- Para valores de la relación SNR altos, mayores que 14 dB, los valores medidos son claramente superiores a los teóricos.

La explicación que cabe argumentar es que la ecuación teórica no tiene en cuenta el efecto de la interferencia intersimbólica (ISI, *Intersymbol Interference*) introducida por el filtro gaussiano. En cualquier caso, la tasa BER que corresponde a este nivel de ruido tiene un impacto poco significativo sobre el retardo. De hecho, se ha podido constatar que, para un nivel de atenuación de ruido de 8 dB, correspondiente a una relación SNR de 14,95 dB, el retardo es prácticamente indistinguible del caso ideal (sin retransmisiones).

- Para valores de SNR inferiores a 10 dB, los valores de la tasa BER medidos están por debajo de la curva correspondiente al mayor índice de modulación ($h = 0,35$).

El hecho de que la tasa BER obtenida experimentalmente infravalore la tasa real, probablemente se deba a que en la medida no se computen todos los paquetes

perdidos, en concreto aquellos que no han sido recibidos a consecuencia del elevado nivel de ruido. No obstante, esta infravaloración no tiene consecuencias, ya que el excesivo nivel de ruido acabaría provocando en un escenario real la pérdida de la conexión, hecho que impide llevar a término medidas de retardo completas.

- **Modulaciones EDR ($\pi/4$ DQPSK y 8 DPSK):** Las tasas BER previstas teóricamente se ajustan en gran medida a las obtenidas de forma empírica.

Conclusiones sobre la validación del retardo teórico

En lo que respecta a la comparación del retardo medido con el calculado teóricamente a partir de la tasa BER medida, se concluye que:

- **Para el modo básico (paquetes DM):**
 - El efecto del intervalo T_{poll} configurado es el mismo que el descrito en el caso en el que no tienen lugar retransmisiones.
 - En el sentido de transmisión esclavo-maestro el maestro no confirma la recepción de un paquete de 5 *slots* que transporta un fragmento inicial de una trama L2CAP y no único, en el *slot* inmediatamente posterior, llegándose a introducir un retardo adicional de hasta 4 *slots*.
 - La separación entre retransmisiones consecutivas resulta tener un valor fijo independiente del sentido de transmisión y del intervalo de T_{poll} configurado: $16 \cdot T_S$, que se suman al tiempo de transmisión del paquete, incluyendo el *slot* de confirmación.
 - El modelo, por lo demás, ajusta el retardo medido con un error relativo medio inferior al 6%.
- **Para el modo EDR:** Aunque se fuerza el uso de paquetes EDR excluyendo la utilización de paquetes DM, el dispositivo, en respuesta al nivel de pérdidas, pasa a emplear los paquetes DM. Este hecho se ha podido constatar a partir del resultado de la medida, ya que la curva del retardo presenta una pendiente de crecimiento del mismo orden que la que corresponde al uso de paquetes DM.

7.2. Líneas futuras

Las líneas que se proponen para continuar investigando son las siguientes:

- Análisis teórico de la inclusión de formatos de paquete no definidos en el estándar Bluetooth: 2-DM y 3-DM.
- Análisis del origen del retardo introducido por la capa de transporte USB.
- Extensión del modelo para caracterizar el funcionamiento de *piconets* y *scatternets* utilizando el perfil SPP y teniendo en cuenta el efecto de la capa de transporte.
- Empleo de otros modelos de canal alternativos al gaussiano. Concretamente se propone el estudio del canal de tipo Rice con desvanecimientos lentos.
- Ampliaciones específicas del modelo a otros perfiles, como PAN (*Personal Area Networks*), que no han sido abordadas: Estudiar el rendimiento, en términos de retardo extremo a extremo, cuando las transmisiones se realizan:
 - Dentro de una misma *piconet*, pero entre dos dispositivos que actúan como esclavos.
 - En una *scatternet*.
 - Entre un esclavo de la red PAN y un equipo de una red IP, a la que tenga acceso el coordinador de la PAN, dotado con la funcionalidad de punto de acceso a otras redes: Ethernet, 802.11 y GPRS.
- Extensión para modelar las versiones Bluetooth 3.0 + HS y Bluetooth 4.0.

7.3. Publicaciones derivadas del desarrollo de la tesis

7.3.1. Revistas incluidas en el JCR

- [1] LUQUE, R., M. MORÓN, y E. CASILARI: «Minimum transmission delay in Bluetooth 2.0 + EDR», *Electronics Letters*, tomo 46, nº 13, jun. 2010, págs. 955–957.
- [2] MORÓN, M., R. LUQUE, E. CASILARI, y A. DÍAZ-ESTRELLA: «Characterization of Bluetooth packet delay in noisy environments», *IEEE Communications Letters*, tomo 13, nº 9, sep. 2009, págs. 661–663.
- [3] MORÓN, M., R. LUQUE, E. CASILARI, y A. DÍAZ-ESTRELLA: «Overhead and Segmentation Mismatch Effect on Bluetooth WPAN Performance», *Wireless Personal Communications*, tomo 50, nº 2, jul. 2009, págs. 161–180.
- [4] MORÓN, M., R. LUQUE, E. CASILARI, y A. DÍAZ-ESTRELLA: «Minimum delay bound in Bluetooth transmissions with serial port profile», *Electronics Letters*, tomo 44, nº 18, ago. 2008, págs. 1099–1100.
- [5] MORÓN, M., R. LUQUE, E. CASILARI, y A. DÍAZ-ESTRELLA: «An Analytical Study of the Delay in Bluetooth Networks Using the Personal Area Network Profile», *IEEE Communications Letters*, tomo 11, nº 11, nov. 2007, págs. 845–847.

7.3.2. Congresos Internacionales

- [6] LUQUE, R., M. J. MORÓN, y E. CASILARI: «Analytical characterisation of the performance of Bluetooth piconets using serial port profile», en *IWCMC '10: Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, ACM, Caen, France, 28 jun. - 2 jul. 2010, págs. 356–360.
- [7] MORÓN, M. J., R. LUQUE, E. CASILARI, y A. DÍAZ-ESTRELLA: «An analytical model for estimating the delay in Bluetooth communications with serial port profile», en *IWCMC '09: Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing*, ACM, Leipzig, Germany, 21-24 jun. 2009, págs. 1178–1183.

- [8] MORÓN, M., R. LUQUE, E. CASILARI, y A. DÍAZ-ESTRELLA: «Analysis of Bluetooth transmission delay in personal area networks», en *3rd International Symposium on Wireless Pervasive Computing, 2008. ISWPC 2008.*, Santorini, Greece, 7-9 sep. 2008, págs. 620–622.

7.3.3. Otras publicaciones

- [9] MORÓN, M. J., J. R. LUQUE, A. GÓMEZ-JAIME, E. CASILARI, y A. DÍAZ-ESTRELLA: «Prototyping of a remote monitoring system for a medical Personal Area Network using Python», en *3rd International Conference on Pervasive Computing Technologies for Healthcare, 2009. PervasiveHealth 2009.*, London, UK, 1-3 abr. 2009, págs. 1 –5.
- [10] MORÓN, M. J., J. R. LUQUE, A. A. BOTELLA, E. J. CUBEROS, E. GALLARDO, E. CASILARI, A. DÍAZ-ESTRELLA, y J. A. GÁZQUEZ: «Development of wireless body area network based on J2ME for m-health applications», en *ECC'08: Proceedings of the 2nd conference on European Computing Conference*, World Scientific and Engineering Academy and Society (WSEAS), Malta, 11-13 sep. 2008, págs. 83–88.
- [11] MORÓN, M., J. LUQUE, A. BOTELLA, E. CUBEROS, E. CASILARI, y A. DÍAZ-ESTRELLA: «A Smart Phone-based Personal Area Network for Remote Monitoring of Biosignals», en *4th International Workshop on Wearable and Implantable Body Sensor Networks*, tomo 13 de *IFMBE Proceedings*, cap. 3rd Session, Springer Berlin Heidelberg, Aachen University, Germany, 2007, págs. 116–121.
- [12] MORÓN, M., J. LUQUE, A. BOTELLA, E. CUBEROS, E. CASILARI, y A. DÍAZ-ESTRELLA: «J2ME and smart phones as platform for a Bluetooth Body Area Network for Patient-telemonitoring», en *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007. EMBS 2007.*, Lyon, France, 22-26 ago. 2007, págs. 2791–2794.
- [13] MORÓN, M., E. CASILARI, R. LUQUE, y J. GÁZQUEZ: «A wireless monitoring system for pulse-oximetry sensors», en *Proceedings of Systems Communications 2005. ICW 2005, Wireless Technologies*, Montreal, Canada, 14-17 ago. 2005, págs. 79 – 84.

- [14] MORÓN, M., J. LUQUE, E. CASILARI, J. GÁZQUEZ, y A. DÍAZ-ESTRELLA: «Monitorización inteligente de pacientes mediante tecnologías inalámbricas», en *Actas del I Simposio sobre Computación Ubicua e Inteligencia Ambiental (UCAMI 2005)*, Granada, sep. 2005, págs. 313–320.
- [15] MORÓN, M., J. LUQUE, y A. DÍAZ-ESTRELLA: «SICILIA: Sistema de Interfaz Corporal Inalámbrico para un entorno de Inteligencia Ambiental», en *Actas de INTERACCIÓN 2003: IV Congreso Internacional de Interacción Persona-Ordenador [recurso electrónico]*, AIPO, Vigo, 2003.

APÉNDICE A

Medidas realizadas con el S.O. Linux Ubuntu 9.10 Karmic Koala

En este anexo se incluyen ejemplos del retardo medido empleando el S.O. Linux Ubuntu 9.10 Karmic Koala, publicado en Octubre del año 2009, con el fin de mostrar la divergencia en el comportamiento respecto a sus antecesoras: Ubuntu Linux 7.04 [Linux] con *kernel* de la versión 2.6.20-*lowlatency* y Ubuntu Linux 8.04 [Linux] con *kernel* de la versión 2.6.24-*rt*.

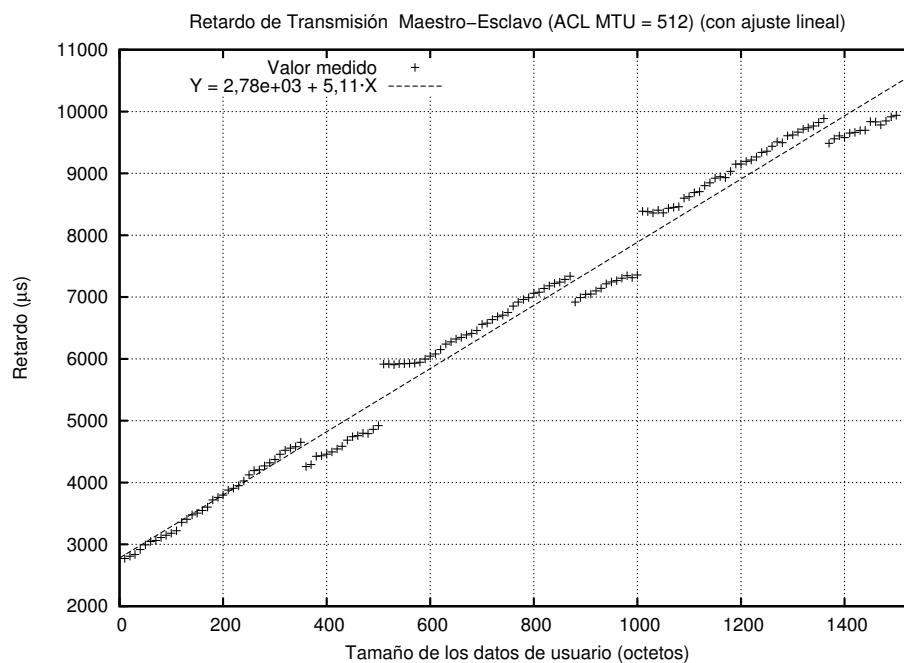
En las figuras A.1 y A.2 se comparan los retardos obtenidos con versiones anteriores (véanse las figuras A.1a y A.2a) cuando el parámetro ACL MTU toma los valores de 512 y 1021 octetos, con los medidos sobre la última versión disponible en el momento de la redacción de la presente tesis, Linux Ubuntu 9.10 Karmic Koala (véanse las figuras A.1b y A.2b). Puede observarse que, para ambos casos, tanto el retardo correspondiente a la mínima longitud medida, como las pendientes de crecimiento son significativamente mayores. Concretamente, cuando el parámetro ACL MTU se configura a un valor de:

- 512 octetos: En la figura A.1a, puede comprobarse que una longitud de 10 octetos tiene asociado un retardo inferior a los 3 ms — $2772\mu s$ —, valor que pasa a ser de $14331\mu s$, cuando se cambia a la versión Linux Ubuntu 9.10 (véase la figura A.1b). Del mismo modo, la transmisión de 1500 octetos pasa de requerir $9938\mu s$ a $30706\mu s$.

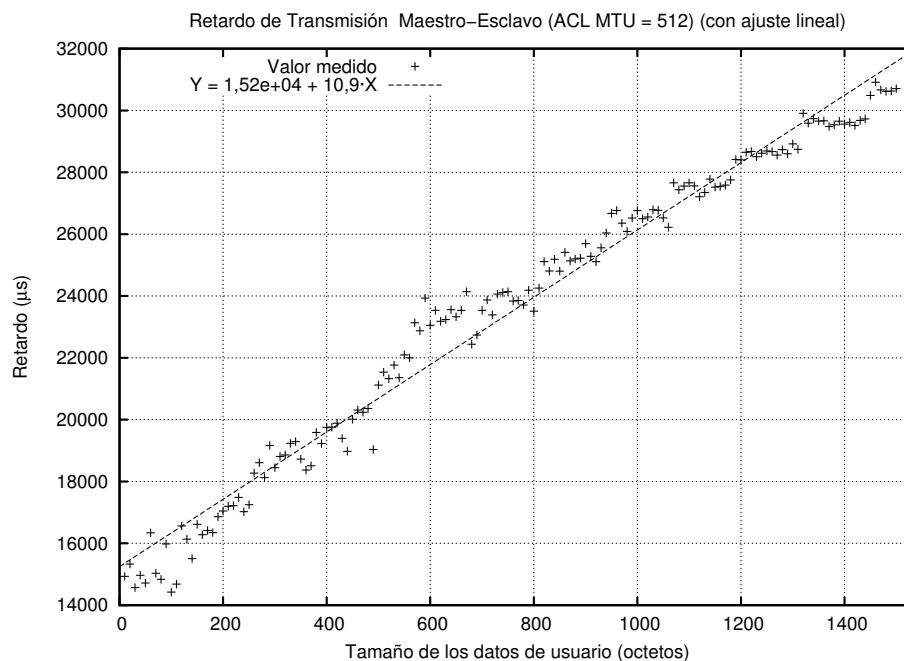
Así, la pendiente, obtenida a partir de los resultados medidos, por el método de mínimos-cuadrados, para el primer caso es de $5,11 \mu\text{s}/\text{byte}$ frente a los $10,9 \mu\text{s}/\text{byte}$ que se alcanzan en el segundo caso, según se observa en las figuras A.1a y A.1b.

- 1021 octetos: A una longitud de 10 octetos corresponde un retardo de $2710 \mu\text{s}$, también inferior a los 3 ms, valor que aumenta hasta los $15054 \mu\text{s}$, con la versión Linux Ubuntu 9.10. Análogamente, en esta misma versión, enviar 1500 octetos supone $31042 \mu\text{s}$, frente a los $9153 \mu\text{s}$ logrados en las versiones anteriores. Según se muestra en las figuras A.2a y A.2b, la pendiente, estimada por el método de mínimos-cuadrados, para el primer caso es de $4,66 \mu\text{s}/\text{byte}$ y de $11,3 \mu\text{s}/\text{byte}$, en el segundo caso.

Adicionalmente se ha constatado que este comportamiento es independiente de la plataforma *hardware* en la que se ejecuta. Con este fin, se ha repetido la medida, configurando el valor del parámetro ACL MTU a un valor de 420 octetos, utilizando dos máquinas muy diferentes —Pentium IV Prescott y Core i7— Para ambos casos se ha obtenido el mismo resultado tal y como puede comprobarse en la figura A.3, con el agravante de que en el primer caso se ha utilizado un *kernel* de tiempo real y en el segundo no.

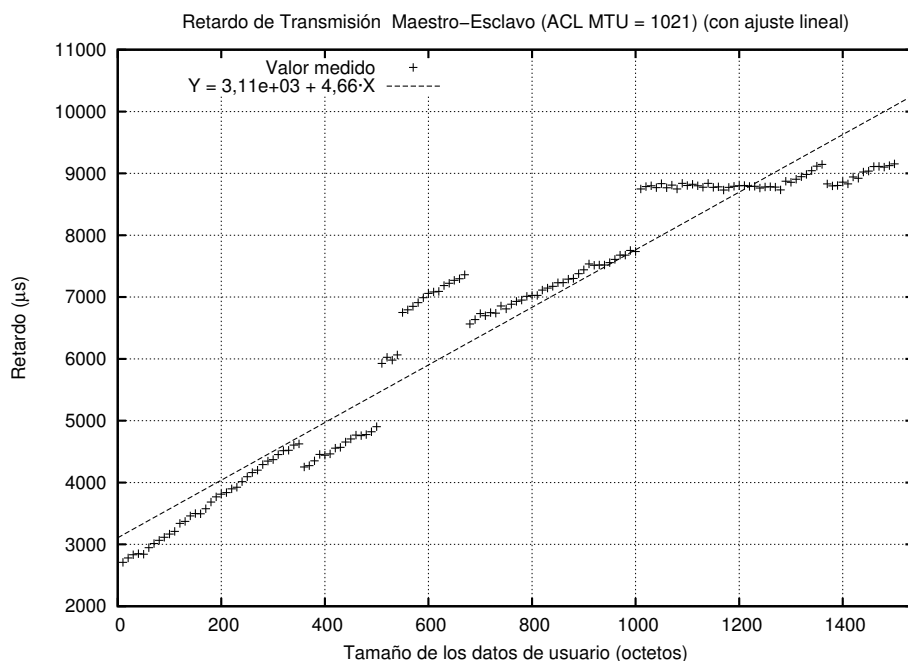


(a) Ubuntu Linux 7.04 con *kernel* de la versión 2.6.20-*lowlatency* y Ubuntu Linux 8.04 con *kernel* de la versión 2.6.24-rt

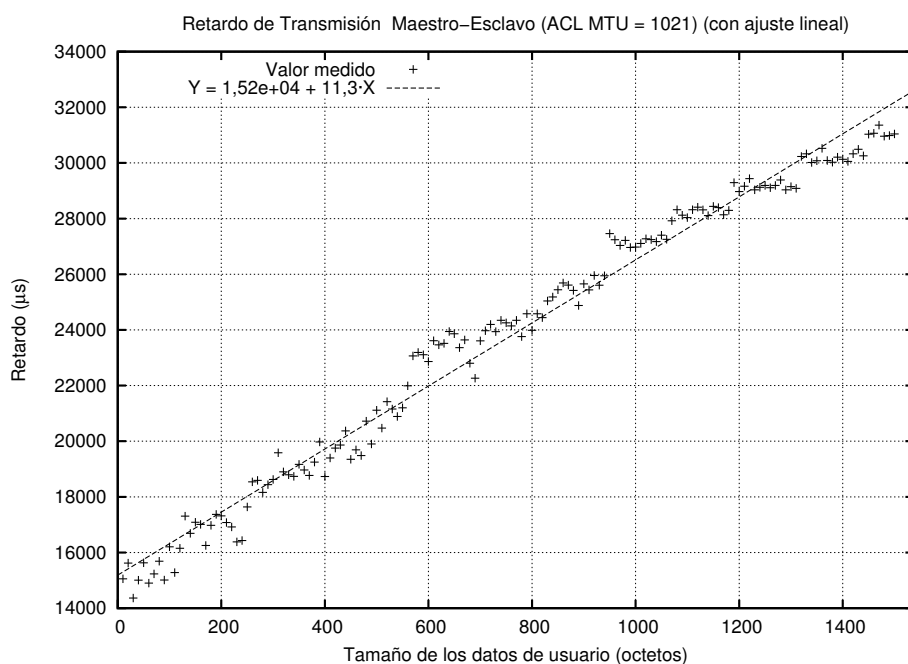


(b) Linux Ubuntu 9.10 Karmic Koala

Figura A.1.: Comparación del retardo obtenido con distintas versiones del S.O. Linux Ubuntu, utilizando dispositivos Bluetooth 2.0 + EDR para un valor del parámetro ACL MTU de 512 octetos (Revisión de HCI: 0xc5c).



(a) Ubuntu Linux 7.04 con *kernel* de la versión 2.6.20-*lowlatency* y Ubuntu Linux 8.04 con *kernel* de la versión 2.6.24-rt



(b) Linux Ubuntu 9.10 Karmic Koala

Figura A.2.: Comparación del retardo obtenido con distintas versiones del S.O. Linux Ubuntu, utilizando dispositivos Bluetooth 2.0 + EDR para un valor del parámetro ACL MTU de 1021 octetos (Revisión de HCI: 0xc5c).

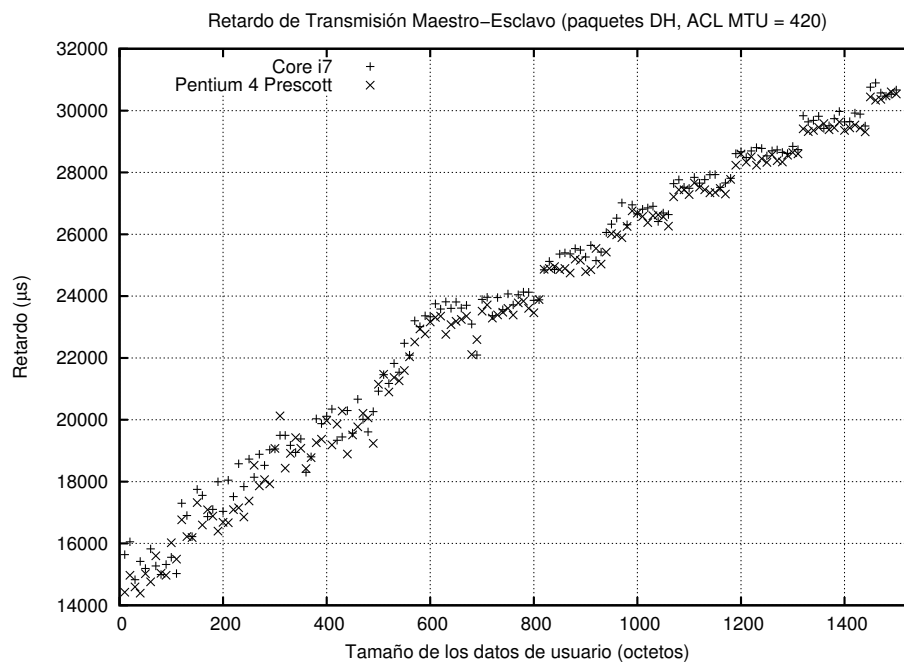


Figura A.3.: Comparación del retardo obtenido, en distintas plataformas de medida, ambas con el S.O. Linux Ubuntu 9.10 Karmic Koala, utilizando dispositivos Bluetooth 2.0 + EDR para un valor del parámetro ACL MTU de 1021 octetos (Revisión de HCI: 0xc5c).

Bibliografía

- [A&D Medical] A&D MEDICAL: «ANDMEDICAL», Internet, [Consulta: 16/06/2010].
URL <http://www.andmedical.com.au/>
- [Agilent Tech.a] AGILENT TECH.: «8494B Manual Step Attenuator, DC to 18 GHz, 0 to 11 dB, 1 dB steps», Internet, [Consulta: 21/06/2010].
URL <http://www.home.agilent.com/agilent/product.jsx?pn=8494B>
- [Agilent Tech.b] AGILENT TECH.: «8496B Manual Step Attenuator, DC to 18 GHz, 0 to 110 dB, 10 dB steps», Internet, [Consulta: 21/06/2010].
URL <http://www.home.agilent.com/agilent/product.jsx?pn=8496B>
- [Agilent Tech.c] AGILENT TECH.: «N9010A EXA Signal Analyzer», Internet, [Consulta: 21/06/2010].
URL <http://www.home.agilent.com/agilent/product.jsx?pn=N9010A>
- [Arumugam02] ARUMUGAM, A., A. NIX, P. FLETCHER, S. ARMOUR, y B. LEE: «Scenario driven evaluation and interference mitigation proposals for Bluetooth and high data rate Bluetooth enabled consumer electronic devices», *IEEE Transactions on Consumer Electronics*, tomo 48, nº 3, ago. 2002, págs. 754 – 764.

- [Arumugam03] ARUMUGAM, A., A. DOUFEXI, A. NIX, y P. FLETCHER: «An investigation of the coexistence of 802.11g WLAN and high data rate Bluetooth enabled consumer electronic devices in indoor home and office environments», *IEEE Transactions on Consumer Electronics*, tomo 49, n^o 3, aug. 2003, págs. 587 – 596.
- [Bamahdi06] BAMAHDİ, O., y S. ZUMMO: «An Adaptive Frequency Hopping Technique- With Application to Bluetooth-WLAN Coexistence», en *International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006. ICN/ICONS/MCL 2006.*, 23-29 abr. 2006, págs. 131 – 131.
- [Benvenuto02] BENVENUTO, N., y G. CHERUBINI: *Algorithms for Communications Systems and their Applications* (Wiley), oct. 2002.
- [BlueGiga Tech.a] BLUEGIGA TECH.: «Bluegiga Technologies», Intenet, [Consulta: 16/06/2010].
URL <http://www.bluegiga.com/>
- [BlueGiga Tech.b] BLUEGIGA TECH.: «WT11 Evaluation Kit», Intenet, [Consulta: 16/06/2010].
URL http://www.bluegiga.com/eval_wt11
- [Bluetooth SIG01] BLUETOOTH SIG: «Bluetooth Core Specification Version 1.1», en *Specification of the Bluetooth System*, tomo 1 (Bluetooth Special Interest Group), 22 feb. 2001.
URL <http://www.bluetooth.com/English/Technology/Building/Pages/Specification.aspx>
- [Bluetooth SIG03] BLUETOOTH SIG: «Bluetooth Core Specification Version 1.2», en *Specification of the Bluetooth System* (Bluetooth Special Interest Group), 5 nov. 2003.
URL <http://www.bluetooth.com/English/Technology/Building/Pages/Specification.aspx>

-
- [Bluetooth SIG04] BLUETOOTH SIG: «Bluetooth Core Specification Version 2.0+EDR», en *Specification of the Bluetooth System* (Bluetooth Special Interest Group), 10 nov. 2004.
URL <http://www.bluetooth.com/English/Technology/Building/Pages/Specification.aspx>
- [Bluetooth SIG07] BLUETOOTH SIG: «Bluetooth Core Specification Version 2.1+EDR», en *Specification of the Bluetooth System* (Bluetooth Special Interest Group), 26 jul. 2007.
URL <http://www.bluetooth.com/English/Technology/Building/Pages/Specification.aspx>
- [Bluetooth SIG09] BLUETOOTH SIG: «Bluetooth Core Specification Version 3.0+HS», en *Specification of the Bluetooth System* (Bluetooth Special Interest Group), 21 abr. 2009.
URL <http://www.bluetooth.com/English/Technology/Building/Pages/Specification.aspx>
- [Bluetooth SIG10] BLUETOOTH SIG: «Bluetooth Core Specification Version 4.0», en *Specification of the Bluetooth System* (Bluetooth Special Interest Group), 30 jun. 2010.
URL <http://www.bluetooth.com/English/Technology/Building/Pages/Specification.aspx>
- [BlueZ] BLUEZ: «BlueZ Project: BlueZ, Bluetooth Protocol Stack for Linux», Internet, [Consulta: 16/01/2010].
URL <http://www.bluez.org/>
- [Cabral09] CABRAL, L., y R. D. LINS: «An analysis of the QoS in the transmission in 802.11g networks in the presence of Bluetooth interference», en *5th International Conference on Wireless and Mobile Communications, ICWMC 2009* (Affiliation: Universidade Federal de Pernambuco, Recife, Brazil; Correspondence Address: Cabral, L.; Universidade Federal de Pernambuco, Recife, Brazil; email: leo.cabral@click21.com.br), 23 August 2009 through 29 August 2009 2009, págs. 76–81.

- [Carlson86] CARLSON, A. B.: *Communication Systems: An Introduction to Signals and Noise in Electrical Communication* (McGraw-Hill Education), tercera edición, 1986.
- [Chandra08] CHANDRA, A., S. C. MANDAL, y C. BOSE: «BER of $\pi/4$ -DQPSK with multichannel reception: Some series solutions», en *TENCON 2008 - 2008 IEEE Region 10 Conference*, 2008, págs. 1–6.
- [Chang06] CHANG, D.-C., y T.-H. SHIU: «Digital GFSK Carrier Synchronization», en *IEEE Asia Pacific Conference on Circuits and Systems, 2006. APCCAS 2006*, 2006, págs. 1523–1526.
- [Chang09] CHANG, D.-C.: «Least Squares/Maximum Likelihood Methods for the Decision-Aided GFSK Receiver», *IEEE Signal Processing Letters*, tomo 16, nº 6, 2009, págs. 517–520.
- [Chen04a] CHEN, L. J., R. KAPOOR, K. LEE, M. Y. SANADIDI, y M. GERLA: «Audio streaming over Bluetooth: an adaptive ARQ timeout approach», en *24th International Conference on Distributed Computing Systems Workshops, 2004*, 2004, págs. 196–201.
- [Chen04b] CHEN, L.-J., R. KAPOOR, M. Y. SANADIDI, y M. GERLA: «Enhancing Bluetooth TCP throughput via link layer packet adaptation», en *Communications, 2004 IEEE International Conference on*, tomo 7, 2004, págs. 4012–4016.
- [Chen04c] CHEN, L.-J., T. SUN, M. SANADIDI, y M. GERLA: «Improving wireless link throughput via interleaved FEC», en *Ninth International Symposium on Computers and Communications, 2004. ISCC 2004*, tomo 1, 28 jun. - 1 jul. 2004, págs. 539– 544.
- [Chen06] CHEN, L.-J., T. SUN, y Y.-C. CHEN: «Improving Bluetooth EDR data throughput using FEC and interleaving», en *Mobile Ad-hoc and Sensor Networks*, (eds.) J. CAO, I. STOJMENOVIC, X. JIA, y S. DAS, tomo 4325/2006 de *Lecture Notes In Computer Science* (Springer-Verlag Berlin), 2006, págs. 724–735.

-
- [Chow94] CHOW, Y. C., A. R. NIX, y J. P. MCGEEHAN: «An error bound analysis for M-DPSK in frequency-selective Rayleigh fading channel with diversity reception», en *Singapore ICCS '94. Conference Proceedings.*, tomo 2, 1994, págs. 686–690.
- [Chui02] CHUI, T. Y., F. THALER, y W. G. SCANLON: «A novel channel modeling technique for performance analysis of Bluetooth baseband packets», en *2002 International Conference on Communications (ICC 2002)*, tomo 1, 28 abr. - 2 mayo 2002, págs. 308–312.
- [Conti03] CONTI, A., D. DARDARI, G. PASOLINI, y O. ANDRISANO: «Bluetooth and IEEE 802.11b coexistence: analytical performance evaluation in fading channels», *IEEE Journal on Selected Areas in Communications*, tomo 21, nº 2, feb 2003, págs. 259 – 269.
- [Cordeiro04] CORDEIRO, C. D. M., S. ABHYANKAR, R. TOSHIWAL, y D. P. AGRAWAL: «BlueStar: enabling efficient integration between bluetooth WPANs and IEEE 802.11 WLANs», *Mobile Networks and Applications*, tomo 9, nº 4, 2004, págs. 409–422.
- [CorScience] CORSCIENCE: «CorScience - cardiovascular innovations - cardiology - wireless sensors - defibrillator, pulse oximeter - Research & Development», Internet, [Consulta: 16/06/2010].
URL <http://www.corscience.de/en/corscience/home.html>
- [CSR] CSR: «Cambridge Silicon Radio Plc. BlueCore Bluetooth chipset.», Internet, [Consulta: 16/06/2010].
URL <http://www.csr.com/products/bcrange.htm>
- [CSR00] CSR: «CSR demonstrates single chip Bluetooth radio at CeBIT», <http://www.csr.com/pr/pr3.htm>, 2000, [Consulta: 20/06/2010].
URL <http://www.csr.com/pr/pr3.htm>
- [CSR10] CSR: «CSR - Better Connected: The Casira development system for Bluetooth», <http://www.csr.com/development/casira.htm>, 2010.
URL <http://www.csr.com/development/casira.htm>

- [El-Bendary09a] EL-BENDARY, M., A. ABU EL-AZM, N. EL-FISHAWY, M. EL-TOKHY, y F. SHAWKY: «Improving power efficiency of Bluetooth systems with EDR packets and efficient channel coding», en *International Conference on Computer Engineering Systems, 2009. ICCES 2009* (Cairo, Egypt), 14-16 ene. 2009, págs. 379–384.
- [El-Bendary09b] EL-BENDARY, M. A. M. M., A. E. A. EL-AZM, N. A. EL-FISHAWY, A. HAGAGG, y F. SHAWKY: «Throughput improvement over Bluetooth system through adaptive packets», en *International Conference on Computer Engineering & Systems, 2009. ICCES 2009*, 2009, págs. 616–621.
- [Electronics07] ELECTRONICS: «CSR and Broadcom Captured over Three Quarters of the Worldwide Bluetooth Semiconductor Market in 2007», Internet, 2007, [Consulta: 20/06/2010].
URL <http://www.electronics.ca/presscenter/articles/874/1/CSR-and-Broadcom-Captured-over-Three-Quarters-of-the-Worldwide-Bluetooth-Semiconductor-Market-in-2007/Page1.html>
- [ETSI97] ETSI: *ETSI TS 101 369:Digital cellular telecommunications system (Phase 2+); Terminal Equipment to Mobile Station (TE-MS) multiplexer protocol (3PP TS 07.10 version 6.5.0 Release 1997)*, ETSI, 1997.
URL <http://www.etsi.org/>
- [Falujah08] FALUJAH, I. A., y V. K. PRABHU: «Error Performance of DQPSK with EGC Diversity Reception over Fading Channels», *IEEE Transactions on Wireless Communications*, tomo 7, n.º 4, 2008, págs. 1190–1194.
- [Feng09] FENG, G., X. QIMIN, y X. QILI: «Bluetooth channel quality simulation, estimation and adaptive packet selection strategy», *WSEAS Transactions on Information Science and Applications*, tomo 6, n.º 1, 2009, págs. 84–94.
- [Finch05] FINCH, S.: «Bluetooth expands into non-consumer markets», *Incisor*, tomo 85, ago. 2005, págs. 8–9.

-
- [Francia04] FRANCIA, G. A., A. KILARU, L. PHUONG, y M. VASHI: «An empirical study of Bluetooth performance», en *MSCCC '04: Proceedings of the 2nd annual conference on Mid-south college computing* (Mid-South College Computing Conference, Little Rock, Arkansas, United States), 2004, págs. 81–93.
- [García-Armada07] GARCÍA-ARMADA, A., B. B. RODRÍGUEZ, V. P. G. JIMÉNEZ, y M. SÁNCHEZ-FERNÁNDEZ: «Modelling, performance analysis and design of WPAN systems», *Wireless Personal Communications*, tomo 42, nº 3, 2007, págs. 367–386.
- [Gnuplot] GNUPLOT: «Gnuplot Homepage», Internet, [Consulta: 16/06/2010].
URL <http://www.gnuplot.info/>
- [Goldsmith05] GOLDSMITH, A.: *Wireless Communications* (Cambridge University Press), ago. 2005.
- [Golmie01a] GOLMIE, N., y N. CHEVROLLIER: «Techniques to improve Bluetooth performance in interference environments», en *IEEE Military Communications Conference, 2001. MILCOM 2001*, IEEE; ComSoc; AFCEA; EDS; MITRE (IEEE, Mclean, VA), 28-31 oct. 2001, págs. 581–585.
- [Golmie01b] GOLMIE, N., N. CHEVROLLIER, y I. ELBAKKOURI: «Interference aware Bluetooth packet scheduling», en *IEEE Global Telecommunications Conference. GLOBECOM'01* (San Antonio, TX), 25-29 nov. 2001, págs. 2857–2863.
- [Golmie01c] GOLMIE, N., y F. MOUVEAUX: «Interference in the 2.4 GHz ISM band: Impact on the Bluetooth access control performance», en *IEEE International Conference On Communications, 2001* (Helsinki, Finland), 11-14 jul. 2001, págs. 2540–2545.
- [Golmie02] GOLMIE, N.: «Performance evaluation of a Bluetooth channel estimation algorithm», en *13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications* (Lisbon, Portugal), 15-18 sep. 2002, págs. 156–160.

- [Golmie03a] GOLMIE, N., N. CHEVROLLIER, y O. REBALA: «Bluetooth and WLAN coexistence: Challenges and solutions», *IEEE Wireless Communications Magazine*, tomo 10, nº 6, dic. 2003, págs. 22–29.
- [Golmie03b] GOLMIE, N., y O. REBALA: «Bluetooth adaptive techniques to mitigate interference», en *IEEE Global Telecommunications Conference. GLOBE-COM'03* (San Francisco, CA), 01-05 dic. 2003, págs. 405–409.
- [Golmie03c] GOLMIE, N., y O. REBALA: «Techniques to improve the performance of TCP in a mixed Bluetooth and WLAN environment», en *IEEE International Conference On Communications 2003* (Anchorage, AK, USA), 11-15 mayo 2003, págs. 1181–1185.
- [Golmie03d] GOLMIE, N., O. REBALA, y N. CHEVROLLIER: «Bluetooth adaptive frequency hopping and scheduling», en *IEEE Military Communications Conference 2003. MILCOM 2003* (Boston, MA), 13-16 oct. 2003, págs. 1138–1142.
- [Golmie03e] GOLMIE, N., R. VAN DYCK, A. SOLTANIAN, A. TONNERRE, y O. REBALA: «Interference evaluation of Bluetooth and IEEE 802.11b systems», *Wireless Networks*, tomo 9, nº 3, mayo 2003, págs. 201–211.
- [Golmie04] GOLMIE, N.: «Bluetooth dynamic scheduling and interference mitigation», *Mobile Networks and Applications*, tomo 9, nº 1, feb. 2004, págs. 21–31.
- [Gopalpet07] GOPALPET, D., y M. SONG: «Performance analysis of 802.11b networks in the presence of interference-aware bluetooth devices», en *4th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness and Workshops, QSHINE '07*, 14 August 2009 through 17 August 2009 2007.
- [Han08] HAN, D., y Y. ZHENG: «An ultra low power GFSK demodulator for wireless body area network», en *34th European Solid-State Circuits Conference, 2008. ESSCIRC 2008*, 15-19 sep. 2008, págs. 434 –437.
- [Han09] HAN, U. D., y Y. ZHENG: «A GFSK demodulator based on instant phase computation and adaptive multi-threshold quantization», en *2009 IEEE*

Asian Solid-State Circuits Conference, A-SSCC 2009 (Singapore), 16-18 nov. 2009, págs. 249–252.

- [Hipólito09] HIPÓLITO, J. I. N., N. C. ARBALLO, J. A. MICHEL-MACARTY, y E. J. GARCIA: «Bluetooth performance analysis in wireless personal area networks», en *Electronics Robotics and Automotive Mechanics Conference. CERMA 2009* (Cuernavaca, Morelos, Mexico), 22-25 sep. 2009, págs. 38–43.
- [Howitt03] HOWITT, I., y S. Y. HAM: «Site specific WLAN and WPAN coexistence evaluation», en *IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, tomo 3, 20 mar. 2003, págs. 1487 –1491.
- [Hsu06] HSU, A.-C., D. WEI, y C.-C. KUO: «Coexistence Mechanism Using Dynamic Fragmentation for Interference Mitigation between Wi-Fi Bluetooth», en *IEEE Military Communications Conference, 2006. MILCOM 2006.*, 23-25 oct. 2006, págs. 1 –7.
- [Hsu10] HSU, C.-F., y C.-Y. LIU: «An Adaptive Traffic-Aware Polling and Scheduling Algorithm for Bluetooth Piconets», *IEEE Transactions on Vehicular Technology*, tomo 59, nº 3, mar. 2010, págs. 1402 –1414.
- [Hua08] HUA, Y., y Y. ZOU: «Analysis of the packet transferring in L2CAP layer of Bluetooth v2.x+EDR», en *International Conference on Information and Automation, 2008. ICIA 2008* (Zhangjiajie, China), 20-23 jun. 2008, págs. 753 –758.
- [Hyun04] HYUN, S. B., G. Y. TAK, S. H. KIM, B. J. KIM, J. KO, y S. S. PARK: «A dual-mode 2.4-Ghz CMOS transceiver for high-rate Bluetooth systems», *ETRI Journal*, tomo 26, nº 3, 2004, págs. 229–240.
- [Ibrahim07] IBRAHIM, N., L. LAMPE, y R. SCHÖBER: «Bluetooth Receiver Design Based on Laurent's Decomposition», *IEEE Transactions on Vehicular Technology*, tomo 56, nº 4, jul. 2007, págs. 1856 –1862.
- [IEEE 802.15 02] IEEE 802.15 : *IEEE Std 802.15.1 - 2002 IEEE Standard for Information technology – Telecommunications and information exchange between*

- systems – Local and metropolitan area networks – Specific requirements Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)*, IEEE, 2002.
URL <http://ieeexplore.ieee.org/servlet/opac?punumber=7932>
- [IEEE 802.15 05] IEEE 802.15 : *IEEE Std 802.15.1 - 2005 IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)*, IEEE, 2005.
URL <http://ieeexplore.ieee.org/servlet/opac?punumber=9980>
- [IETF-NWG92] IETF-NWG: *RFC 1363: A Proposed Flow Specification*, Internet Engineering Task Force - Network Working Group, 1992.
URL <http://www.ietf.org/rfc/rfc1363.txt?number=1363>
- [Issac09] ISSAC, B.: «Study of mixed mode scenarios with IEEE 802.11 networks and Bluetooth network», en *IEEE Symposium on Industrial Electronics Applications, 2009. ISIEA 2009.*, tomo 1 (Kuala Lumpur, Malaysia), 4-6 oct. 2009, págs. 319–324.
- [ISSC] ISSC: «Integrated System Solution Corp. (I.S.S.C.)», Intenet, [Consulta: 16/06/2010].
URL <http://www.issc.com.tw/>
- [ITU-T96] ITU-T: *ITU-T Recommendation O.150*, International Telecommunication Union, 1996.
- [IxChariot] IxCHARIOT: «Qcheck - Free Network Benchmark Utility IxChariot - Test Device and System Performance under Real-World Load Conditions», [Consulta: 19/07/2010].
URL <http://www.ixchariot.com/products/datasheets/qcheck.html>

-
- [Jiang03] JIANG, B., y O. YANG: «Performance evaluation of Bluetooth system in the presence of WLAN IEEE802.11 system», en *IEEE Canadian Conference on Electrical and Computer Engineering, 2003. CCECE 2003*, tomo 3, 2003, págs. 1633–1636.
- [Jo03] JO, J.-H., y H. JAYANT: «Performance evaluation of multiple IEEE 802.11b WLAN stations in the presence of Bluetooth radio interference», en *IEEE International Conference on Communications, 2003. ICC '03*, tomo 2, 2003, págs. 1163–1168.
- [Kahveci10] KAHVECI, S., y I. KAYA: «Turbo Code with Modified LLR and Performance Analysis for Bluetooth Packets in Indoor Environments», *Wireless Personal Communications*, tomo 54, nº 2, jul. 2010, págs. 277–287.
- [Kalvala06] KALVALA, J., y G. ZARUBA: «Differentiated Priority Scheduling and Adaptive Segmentation for Bluetooth Piconets», en *Proceedings of the 39th Annual Hawaii International Conference on System Sciences, 2006. HICSS'06*, tomo 9, 04-07 ene. 2006, págs. 233b – 233b.
- [Kao07] KAO, H. S., M. J. YANG, y T. C. LEE: «A delay-line-based GFSK demodulator for low-IF receivers», en *54th IEEE International Solid-State Circuits Conference, ISSCC 2007*, 11-15 feb. 2007.
- [Karjalainen02] KARJALAINEN, O., S. RANTALA, y M. KIVIKOSKI: «The performance of Bluetooth system in the presence of WLAN interference in an office environment», en *The 8th International Conference on Communication Systems, 2002. ICCS 2002*, tomo 2, 2002, págs. 628–631.
- [Kiasaleh96] KIASALEH, K., y T. HE: «Performance of DQPSK communication systems impaired by mixer imbalance, timing error, and Rayleigh fading», en *IEEE International Conference on Communications, 1996. ICC 96*, tomo 1, 1996, págs. 364–368.
- [Kiasaleh97] KIASALEH, K., y T. HE: «On the performance of DQPSK communication systems impaired by timing error, mixer imbalance, and frequency nonselective slow Rayleigh fading», *IEEE Transactions on Vehicular Technology*, tomo 46, nº 3, 1997, págs. 642–652.

- [Kostic08] KOSTIC, I.: «Average symbol error probability for M-ary DDPSK», *Electronics Letters*, tomo 44, nº 20, 2008, págs. 1203–1204.
- [Lee06] LEE, T. C., y C. C. CHEN: «A mixed-signal GFSK demodulator for Bluetooth», *IEEE Transactions on Circuits and Systems II: Express Briefs*, tomo 53, nº 3, 2006, págs. 197–201.
- [Lee07] LEE, S., B. PARK, y H. LATCHMAN: «Enhanced Limited Round Robin with a Priority policy for Bluetooth», en *The 9th International Conference on Advanced Communication Technology*, tomo 1, 12-14 feb. 2007, págs. 182 –185.
- [Li07a] LI, J., y X. LIU: «A Frequency Diversity Technique for Interference Mitigation in Coexisting Bluetooth and WLAN», en *IEEE International Conference on Communications, 2007. ICC '07.*, 24-28 jun. 2007, págs. 5490–5495.
- [Li07b] LI, J., y X. LIU: «Evaluation of Co-channel and Adjacent Channel Interference for Multiple Bluetooth Piconets with Dual Channel Transmission», en *IEEE Wireless Communications and Networking Conference, 2007.WCNC 2007.*, 11-15 mar. 2007, págs. 2355 –2360.
- [Li07c] LI, J., X. LIU, y X. MA: «Dual Channel Transmission for Coexistence of Bluetooth Piconets with Multi-Slot Packets», en *2nd International Symposium on Wireless Pervasive Computing, 2007. ISWPC '07.*, 5-7 feb. 2007.
- [Li09] LI, J., X. LIU, y A. SWAMI: «Collision analysis for coexistence of multiple bluetooth piconets and WLAN with dual channel transmission», *IEEE Transactions on Communications*, tomo 57, nº 4, april 2009, págs. 1129 –1139.
- [Linux] LINUX, U.: «Ubuntu Homepage», Intenet, [Consulta: 11/05/2010].
URL <http://www.ubuntu.com/>

-
- [Liu03] LIU, H., V. VENKATESAN, C. NILSEN, R. KYKER, y M. E. MAGANA: «Performance of frequency hopped noncoherent GFSK in correlated Rayleigh fading channels», en *IEEE International Conference on Communications, 2003. ICC '03*, tomo 4, 2003, págs. 2779–2783.
- [Mathew09] MATHEW, A., N. CHANDRABABU, K. ELLEITHY, y S. RIZVI: «IEEE 802.11 - Bluetooth Interference: Simulation and Coexistence», en *Seventh Annual Communication Networks and Services Research Conference, 2009. CNSR '09*. (Moncton, NB), 11-13 mayo 2009, págs. 217 –223.
- [MathWorks] MATHWORKS: «MathWorks - MATLAB and Simulink for Technical Computing», <http://www.mathworks.com/>, [Consulta: 21/06/2010]. URL <http://www.mathworks.com/>
- [Mikulka08] MIKULKA, J., y S. HANUS: «Bluetooth EDR Physical Layer modeling», en *18th International Conference Radioelektronika, 2008* (Prague), 24-25 abr. 2008, págs. 1 –4.
- [Miller98] MILLER, L., y J. LEE: «BER expressions for differentially detected $\pi/4$ DQPSK modulation», *IEEE Transactions on Communications*, tomo 46, nº 1, ene. 1998, págs. 71–81.
- [Mistic02a] MISIC, V., y J. MISIC: «Bluetooth scatternet with a master/slave bridge: a queueing theoretic analysis», en *IEEE Global Telecommunications Conference, 2002. GLOBECOM'02*, tomo 1, 17-21 nov. 2002, págs. 207 – 211.
- [Mistic02b] MISIC, V., y J. MISIC: «Minimizing end-to-end delays in Bluetooth scatternet with a slave/slave bridge», en *Eleventh International Conference on Computer Communications and Networks, 2002*, 14-16 oct. 2002, págs. 634 – 639.
- [Mistic02c] MISIC, V., y J. MISIC: «On Bluetooth piconet traffic performance», en *13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2002. PIMRC 2002*, tomo 1, 15-18 sep. 2002, págs. 501 – 505.

- [Misic02d] MISIC, V., y J. MISIC: «Queueing analysis of delays in Bluetooth slave/slave bridge», en *10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, 2002. MASCOTS 2002*, 2002, págs. 155 – 158.
- [Misic03a] MISIC, J., K. L. CHAN, y V. MISIC: «E-limited scheduling in Bluetooth piconets: performance and admission control», en *2003 IEEE 58th Vehicular Technology Conference, 2003. VTC 2003-Fall*, tomo 5, 6-9 oct. 2003, págs. 2865 – 2869.
- [Misic03b] MISIC, J., y V. MISIC: «Adaptive management of Bluetooth master/slave bridge», en *IEEE International Performance, Computing, and Communications Conference, 2003.*, 9-11 abr. 2003, págs. 199 – 206.
- [Misic03c] MISIC, J., y V. MISIC: «Bridges of Bluetooth county: topologies, scheduling, and performance», *IEEE Journal on Selected Areas in Communications*, tomo 21, nº 2, feb. 2003, págs. 240 – 258.
- [Misic03d] MISIC, J., y V. MISIC: «Modeling Bluetooth piconet performance», *IEEE Communication Letters*, tomo 7, nº 1, ene. 2003, págs. 18 – 20.
- [Misic03e] MISIC, V., E. KO, y J. MISIC: «Adaptive cycle-limited scheduling scheme for Bluetooth piconets», en *14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003*, tomo 2, 7-10 sep. 2003, págs. 1064 – 1068.
- [Misic03f] MISIC, V., y J. MISIC: «Performance of Bluetooth bridges in scatternets with exhaustive service scheduling», en *36th Annual Hawaii International Conference on System Sciences, 2003*, 6-9 ene. 2003, págs. 1–10.
- [Misic03g] MISIC, V., J. MISIC, y K. L. CHAN: «Improving the performance of Bluetooth piconets with synchronous and asynchronous traffic», en *IEEE Global Telecommunications Conference, 2003. GLOBECOM'03*, tomo 1, 1-5 dic. 2003, págs. 205 – 209.

-
- [Misic04a] MISIC, J., K. L. CHAN, y V. MISIC: «Admission control in Bluetooth piconets», *IEEE Transactions on Vehicular Technology*, tomo 53, nº 3, mayo 2004, págs. 890 – 911.
- [Misic04b] MISIC, J., y V. MISIC: «Bluetooth master/slave bridge scheduling with and without rendezvous points», en *24th International Conference on Distributed Computing Systems Workshops, 2004*, 23-24 mar. 2004, págs. 742 – 747.
- [Misic04c] MISIC, J., y V. MISIC: «Performance of Bluetooth piconets with finite buffers», en *First International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, 2004. QSHINE 2004*, 18-20 oct. 2004, págs. 282 – 289.
- [Misic04d] MISIC, J., V. MISIC, y E. KO: «Fixed cycles and adaptive bandwidth allocation can coexist in bluetooth», *Canadian Journal of Electrical and Computer Engineering*, tomo 29, nº 1, ene. - abr. 2004, págs. 135 –147.
- [Misic04e] MISIC, V., E. KO, y J. MISIC: «Load and QoS-adaptive scheduling in Bluetooth piconets», en *37th Annual Hawaii International Conference on System Sciences, 2004*, 5-8 ene. 2004, págs. 1–8.
- [Misic05a] MISIC, J., K. L. CHAN, y V. MISIC: «TCP traffic in Bluetooth 1.2: performance and dimensioning of flow control», en *2005 IEEE Wireless Communications and Networking Conference*, tomo 3, 13-17 mar. 2005, págs. 1798 – 1804.
- [Misic05b] MISIC, J., V. MISIC, y G. REDDY: «On the performance of Bluetooth scatternets with finite buffers», en *25th IEEE International Conference on Distributed Computing Systems Workshops, 2005*, 6-10 jun. 2005, págs. 865–870.
- [Misic06] MISIC, J., y V. B. MISIC: *Performance modeling and analysis of Bluetooth networks : polling, scheduling, and traffic control* (Auerbach Publications. Taylor & Francis Group), 2006.

- [Moon05] MOON, T. K.: *Error Correction Coding: Mathematical Methods and Algorithms* (Wiley-Interscience), jun. 2005.
- [Moon07] MOON, I.-Y.: «Performance Analysis of WAP in Bluetooth Ad-Hoc Network System», en *Knowledge-Based Intelligent Information and Engineering Systems*, tomo 4692/2010 de *Lecture Notes in Computer Science* (Springer-Verlag Berlin / Heidelberg), 2007, págs. 390–396.
- [Moon08] MOON, I. Y.: «Performance analysis of ACL packets using turbo code in Bluetooth wireless system», en *New Directions in Intelligent Interactive Multimedia*, (eds.) T. G.T., V. M.V., H. R.H., y J. L.C.J., tomo 142/2008 de *Studies in Computational Intelligence* (Springer Berlin / Heidelberg), 2008, págs. 313–320.
- [Moron08] MORON, M.: *Estudio del rendimiento de perfiles Bluetooth en Redes de Área Personal*, Tesis Doctoral, E.T.S. Ingenieros de Telecomunicación. Universidad de Málaga, 2 jul. 2008.
- [Navarro-Garcia05] NAVARRO-GARCIA, L., y J. ALONSO: «A novel method for interference analysis in IEEE 802.11 WLAN's in coexistence with Bluetooth», en *The European Conference on Wireless Technology, 2005.*, 3-4 abr. 2005, págs. 181 –184.
- [Ng93] NG, C. S., T. T. TJHUNG, F. ADACHI, y K. M. LYE: «On the error rates of differentially detected narrowband $\pi/4$ -DQPSK in Rayleigh fading and Gaussian noise», *IEEE Transactions on Vehicular Technology*, tomo 42, nº 3, 1993, págs. 259–265.
- [NoiseCom] NOISECOM: «Analog Noise Generators NC6000A/8000A», Internet, [Consulta: 21/07/2010].
URL <http://noisecom.com/en/Products/Instruments/NC6000A8000ASeries.aspx>
- [Nokia] NOKIA: «Nokia Energy Profiler», Internet, [Consulta: 21/06/2010].
URL http://www.forum.nokia.com/info/sw.nokia.com/id/324866e9-0460-4fa4-ac53-01f0c392d40f/Nokia_Energy_Profiler.html

-
- [Noneaker94] NONEAKER, D. L., y M. B. PURSLEY: «Error probability bounds for M-PSK and M-DPSK and selective fading diversity channels», *IEEE Transactions on Vehicular Technology*, tomo 43, nº 4, 1994, págs. 997–1005.
- [Nonin Medical] NONIN MEDICAL: «Nonin Medical», Internet, [Consulta: 21/06/2010].
URL <http://www.nonin.com/>
- [NS-2] NS-2: «The Network Simulator (ns-2)», Internet, [Consulta: 21/06/2010].
URL <http://www.isi.edu/nsnam/ns/>
- [NumPy] NUMPY: «Scientific Computing Tools for Python – NumPy», Internet, [Consulta: 11/06/2010].
URL <http://numpy.scipy.org/>
- [Ophir04] OPHIR, L., Y. BITRAN, y I. SHERMAN: «Wi-Fi (IEEE 802.11) and Bluetooth coexistence: issues and solutions», en *15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004.*, tomo 2, 5-8 sep. 2004, págs. 847– 852.
- [OPNET] OPNET: «OPNET Modeler», Internet, [Consulta: 21/06/2010].
URL http://www.opnet.com/solutions/network_rd/modeler.html
- [Pang06] PANG, Q., y V. LEUNG: «Improved Channel Classification and Scheduling for Non-collaborative Bluetooth/ WLAN Coexistence», en *IEEE 63rd Vehicular Technology Conference, 2006. VTC 2006-Spring.*, tomo 3 (Melbourne, Australia), 7-10 mayo 2006, págs. 1303 –1307.
- [Papoulis02] PAPOULIS, A., y S. UNNIKRISHNA PILLAI: *Probability, Random Variables and Stochastic Processes* (McGraw Hill Higher Education), cuarta ed^{ón}, ene. 2002.
- [Perrucci09] PERRUCCI, G., M. PEDERSEN, T. MADSEN, y F. FITZEK: «Energy evaluation for Bluetooth link layer packet selection scheme», en *European Wireless Conference, 2009. EW 2009* (Aalborg, Denmark), 17-20 mayo 2009, págs. 250–254.

- [Proakis00] PROAKIS, J.: *Digital Communications* (McGraw-Hill Science/Engineering/Math), cuarta ed^{ón}., ago. 2000.
- [Python] PYTHON: «Python Programming Language - Official Website», Internet, [Consulta: 11/06/2010].
URL <http://www.python.org/>
- [Rashid06] RASHID, R. A., y R. YUSOFF: «Bluetooth performance analysis in personal area network (PAN)», en *International RF and Microwave Conference, RFM 2006* (Putra Jaya, Malaysia), 12-14 sep. 2006, págs. 393–397.
- [Reddy08] REDDY, G., y S. BHATNAGAR: «An efficient and optimized bluetooth scheduling algorithm for scatternets», en *2nd International Symposium on Advanced Networks and Telecommunication Systems, 2008. ANTS'08*, 15-17 dic. 2008, págs. 1 –3.
- [Reuters07] REUTERS: «Atheros enters Bluetooth chip market», Internet, 2007, [Consulta: 20/06/2010].
URL <http://www.reuters.com/article/idUSN2918291720070129>
- [Roh06] ROH, J. S.: «Performance Analysis and Evaluation of Bluetooth Networks in Wireless Channel Environment», en *International Conference on Systems and Networks Communications, 2006. ICSNC '06* (Tahiti), oct. 2006, págs. 61–61.
- [Roy07] ROY, R., M. KUMAR, N. SHARMA, y S. SURAL: «Bottom-Up Construction of Bluetooth Topology under a Traffic-Aware Scheduling Scheme», *IEEE Transactions on Mobile Computing*, tomo 6, n^º 1, ene. 2007, págs. 72 –86.
- [Roy10] ROY, M., y H. JAMADAGNI: «Comparative Study of Adaptive Frequency Hopping with Power Control to Avoid WLAN Interference in WPAN Systems like Bluetooth», en *7th IEEE Consumer Communications and Networking Conference (CCNC), 2010* (Las Vegas, NV, USA), 9-12 ene. 2010, págs. 1 –5.

-
- [RSOFT] RSOFT: «RSOFT Design Group - Artifex development platform», Internet, [Consulta: 19/07/2010].
URL <http://www.rsoftdesign.com/products.php?sub=System+and+Network&itm=Artifex>
- [SciPy] SciPy: «Scientific Tools for Python – SciPy», Intenet, [Consulta: 11/06/2010].
URL <http://www.scipy.org/>
- [Sheng03] SHENG, W., B. XIA, A. E. EMIRA, C. XIN, A. Y. VALERO-LÓPEZ, S. T. MOON, y E. SÁNCHEZ-SINENCIO: «A 3-V, 0.35- μ m CMOS bluetooth receiver IC», *IEEE Journal of Solid-State Circuits*, tomo 38, nº 1, 2003, págs. 30–42.
- [Shimizu97] SHIMIZU, M., N. AOKI, K. SHIRAKAWA, Y. TOZAWA, N. OKUBO, y Y. DAIDO: «New method of analyzing BER performance of GFSK with postdetection filtering», *IEEE Transactions on Communications*, tomo 45, nº 4, 1997, págs. 429–436.
- [SIG10] SIG, B.: «Bluetooth Product Directory», Intenet, 21 jul. 2010, [Consulta: 21/07/2010].
URL <http://www.bluetooth.com/English/Products/Pages/Products.aspx>
- [Simon98] SIMON, M., y M. ALOUINI: «A unified approach to the performance analysis of digital communication over generalized fading channels», *Proceedings of the IEEE*, tomo 86, nº 9, sep. 1998, págs. 1860 –1877.
- [Simon05] SIMON, M. K., y M.-S. ALOUINI: *Digital Communication over Fading Channels (Wiley Series in Telecommunications and Signal Processing)* (Wiley-IEEE Press), segunda ed^{ón}., dic. 2005.
- [Skudlarek02] SKUDLAREK, J.: «RadiSim - a fast digital RF behavioral simulator including bit error rate assessment for system exploration, validation, and tuning», en *Proceedings of the 2002 IEEE International Workshop on Behavioral Modeling and Simulation, 2002. BMAS 2002*, 6-8 oct. 2002, págs. 61 – 66.

- [Song07] SONG, M., S. SHETTY, y D. GOPALPET: «Coexistence of IEEE 802.11b and bluetooth: An integrated performance analysis», *Mobile Networks and Applications*, tomo 12, nº 5-6, 2007, págs. 450–459.
- [Spice] SPICE: «The Spice Home Page», Internet, [Consulta: 20/07/2010].
URL <http://bwrc.eecs.berkeley.edu/Classes/IcBook/SPICE/>
- [Stein64] STEIN, S.: «Unified analysis of certain coherent and noncoherent binary communications systems», *IEEE Transactions on Information Theory*, tomo 10, nº 1, 1964, págs. 43–51.
- [Strangman00] STRANGMAN, G.: «Python-Stats», Internet, 2000, [Consulta: 11/06/2010].
URL http://www.nmr.mgh.harvard.edu/Neural_Systems_Group/gary/python.html
- [Sun09] SUN, Y., A. BARICZ, M. ZHAO, X. XU, y S. ZHOU: «Approximate average bit error probability for DQPSK over fading channels», *Electronics Letters*, tomo 45, nº 23, nov. 2009, págs. 1177–1179.
- [Taher09] TAHER, T., K. RELE, y D. ROBERSON: «Development and Quantitative Analysis of an Adaptive Scheme for Bluetooth and Wi-Fi Co-Existence», en *6th IEEE Consumer Communications and Networking Conference, 2009. CCNC 2009.*, 10-13 ene. 2009, págs. 1 –2.
- [Tan09] TAN, T. H., Y. F. HUANG, H. M. CHIU, y C. S. CHANG: «Enhancing bluetooth data throughput using a mode/packet switching scheme based on ARQN statistics», en *2009 International Conference on Machine Learning and Cybernetics*, tomo 5 (Baoding, China), 12-15 jul. 2009, págs. 2916–2921.
- [Tibenderana04] TIBENDERANA, C., y S. WEISS: «Low-complexity high-performance GFSK receiver with carrier frequency offset correction», en *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04)*, tomo 4, 17-21 mayo 2004, págs. 933–936.

-
- [USB.org00] USB.ORG: *Universal Serial Bus Specification Revision 2.0*, USB Implementers Forum, Inc, abr. 2000.
URL <http://www.usb.org/developers/docs/>
- [Valenti02a] VALENTI, M. C., y M. ROBERT: «Improving the QoS of Bluetooth through turbo coding», en *2002 MILCOM Proceedings*, tomo 2, 7-10 oct. 2002, págs. 1057–1061.
- [Valenti02b] VALENTI, M. C., M. ROBERT, y J. H. REED: «On the throughput of Bluetooth data transmissions», en *IEEE Wireless Communications and Networking Conference, 2002. WCNC2002*, tomo 1, 17-21 mar. 2002, págs. 119–123.
- [Vitetta97a] VITETTA, G., U. MENGALI, y D. TAYLOR: «Noncoherent detection of FSK signals transmitted over linearly time-selective Rayleigh fading channels», en *IEEE International Conference On Communications, 1997. ICC'97*, 1997, págs. 904–908.
- [Vitetta97b] VITETTA, G., U. MENGALI, y D. TAYLOR: «Optimal noncoherent detection of FSK signals transmitted over linearly time-selective Rayleigh fading channels», *IEEE Transactions on Communications*, tomo 45, nº 11, nov. 1997, págs. 1417–1425.
- [Vitetta98] VITETTA, G., U. MENGALI, y D. TAYLOR: «Error probability with incoherent diversity reception of FSK signals transmitted over fast Rician fading channels», *IEEE Transactions on Communications*, tomo 46, nº 11, nov. 1998, págs. 1443–1447.
- [Vitetta99a] VITETTA, G., U. MENGALI, y D. TAYLOR: «An error probability formula for noncoherent orthogonal binary FSK with dual diversity on correlated Rician channels», *IEEE Communications Letters*, tomo 3, nº 2, feb. 1999, págs. 43–45.
- [Vitetta99b] VITETTA, G. M., U. MENGALI, y D. P. TAYLOR: «Error probability of FSK incoherent diversity reception with fast rice fading», *International Journal of Wireless Information Networks*, tomo 6, nº 2, 1999, págs. 107–118.

- [Warriach08] WARRIACH, E. U., y S. WITTE: «Approach for performance investigation of different Bluetooth modules and communication modes», en *4th International Conference on Emerging Technologies, 2008. ICET 2008*, 2008, págs. 167–171.
- [Wong92] WONG, D. P. C., y P. T. MATHIOPOULOS: «Nonredundant error correction analysis and evaluation of differentially detected $\pi/4$ -shift DQPSK systems in a combined CCI and AWGN environment», *IEEE Transactions on Vehicular Technology*, tomo 41, nº 1, 1992, págs. 35–48.
- [Xia03] XIA, B., C. XIN, W. SHENG, A. Y. VALERO-LOPEZ, y E. SÁNCHEZ-SINENCIO: «A GFSK demodulator for low-IF Bluetooth receiver», *IEEE Journal of Solid-State Circuits*, tomo 38, nº 8, 2003, págs. 1397–1400.
- [Xie09] XIE, J., y I. HOWITT: «Multi-domain WLAN load balancing in WLAN/WPAN interference environments», *IEEE Transactions on Wireless Communications*, tomo 8, nº 9, sep. 2009, págs. 4884–4894.
- [Xin02] XIN, C., B. XIA, W. SHENG, A. VALERO-LOPEZ, y E. SANCHEZ-SINENCIO: «A mixed-mode IF GFSK demodulator for Bluetooth», en *IEEE International Symposium on Circuits and Systems, 2002. ISCAS 2002*, tomo 3, 2002, págs. 457–460.
- [Xu07] XU, F., y Y. ZHUANG: «A method to enhance the throughput performance of bluetooth 2.0+EDR specification», en *2007 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2007* (Shanghai, China), 21-25 sep. 2007, págs. 2132–2135.
- [Yang10] YANG, Z., y D. ZHAO: «A polling MAC scheme for enhancing qos support in Bluetooth technology», en *2010 International Conference on E-Health Networking, Digital Ecosystems and Technologies (EDT)*, tomo 1, 17-18 abr. 2010, págs. 450–453.
- [Yin09] YIN, L., R. ZHAO, y Z. GAO: «The Reducing of the Bluetooth Interference in IEEE802.11b», en *1st International Conference on Information Science and Engineering (ICISE)*, 2009, 26-28 dic. 2009, págs. 2746–2749.

-
- [Yoon04] YOON, J. H., S. B. LEE, y S. C. PARK: «Packet and modulation type selection scheme based on channel quality estimation for bluetooth evolution systems», *IEEE Wireless Communications and Networking Conference, WCNC 2004*, tomo 2, 21-25 mar. 2004, págs. 1014–1017.
- [Zanella09a] ZANELLA, A.: «A mathematical framework for the performance analysis of bluetooth with enhanced data rate», *IEEE Transactions on Communications*, tomo 57, nº 8, 2009, págs. 2463–2473.
- [Zanella09b] ZANELLA, A.: «Carrier-Sense ARQ: Squeezing Out Bluetooth Performance While Preserving Standard Compliancy», en *IEEE International Conference on Communications, 2009. ICC'09* (Dresden, Germany), 14-18 jun. 2009, págs. 1 –5.
- [Zhihong09] ZHIHONG, Q., L. DAN, G. TINGYAN, y L. TIANCHI: «Downlink Scheduling Algorithm for Bluetooth Piconet and COV Evaluation», en *WRI International Conference on Communications and Mobile Computing, 2009. CMC '09*, tomo 2, 6-8 ene. 2009, págs. 290 –293.
- [Zielinski07] ZIELINSKI, B.: «IEEE 802.11 network behaviour in the presence of Bluetooth network», en *Second International Conference on Systems and Networks Communications, 2007. ICSNC 2007.*, 25-31 ago. 2007, págs. 18 –18.
- [Zussman07] ZUSSMAN, G., A. SEGALL, y U. YECHIALI: «On the Analysis of the Bluetooth Time Division Duplex Mechanism», *IEEE Transactions on Wireless Communications*, tomo 6, nº 6, jun. 2007, págs. 2149 –2161.

