



UNIVERSIDAD DE MÁLAGA
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
DE TELECOMUNICACIÓN

TESIS DOCTORAL

**Estudio del Rendimiento
de Perfiles Bluetooth
en Redes de Área Personal**

María José Morón Fernández

Málaga, 2008

Dr. EDUARDO CASILARI PÉREZ Y Dr. ANTONIO DÍAZ ESTRELLA,
PROFESORES TITULARES DEL DEPARTAMENTO DE TECNOLOGÍA
ELECTRÓNICA DE LA UNIVERSIDAD DE MÁLAGA

INFORMAN:

Que Dña. María José Morón Fernández, Ingeniera 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:

«ESTUDIO DEL RENDIMIENTO DE PERFILES BLUETOOTH EN REDES DE
ÁREA PERSONAL»

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 29 de Abril de 2008

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

Fdo. Antonio Díaz Estrella
Profesor Titular del Dpto. de
Tecnología Electrónica

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

TESIS DOCTORAL

Estudio del Rendimiento de Perfiles Bluetooth en Redes de Área Personal

Autora:

María José Morón Fernández
Ingeniera de Telecomunicación

Directores:

D. Eduardo Casilari Pérez
Doctor Ingeniero de Telecomunicación
D. Antonio Díaz Estrella
Doctor Ingeniero de Telecomunicación

A Rafa y a mis padres

Agradecimientos

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

A Eduardo Casilari Pérez y Antonio Díaz Estrella, directores de esta tesis, por la gran profesionalidad demostrada. Les agradezco el tiempo dedicado y, sobre todo, el interés y el entusiasmo con que han contribuido a la elaboración de esta tesis. Les agradezco la confianza en mí depositada y su templanza. Siempre me acompañaron, aún cuando el desánimo me revelaba un camino largo y sinuoso.

A Antonio Vallecillo Moreno, profesor del departamento de Lenguajes y Ciencias de la Computación, que tuve el privilegio de conocer en las aulas, sorprendiéndome sobremano la jovialidad que imprimía en sus clases. Su entusiasmo, harto contagioso, me ayudó a recobrar la autoestima, y sus palabras de ánimo fueron un aliciente para continuar el camino cuando más desorientada estaba.

A Rafaela Benítez Rochel, profesora del departamento de Lenguajes y Ciencias de la Computación, por escucharme. Gracias por enseñarme que en

la vida todo acaba, por convencerme de que mientras vivimos esos instantes que pesan sobre nosotros como una carga insostenible, por inverosímil que nos resulte, al final la carga se torna exigua, y esos instantes, en pequeños lapsos de nuestras vidas.

A mis padres, por haberme enseñado a luchar y haberme transmitido, con su ejemplo, que el valor de no rendirse curte el espíritu para la lidia del mañana. Gracias por vuestro apoyo incondicional.

A Rafa, por confiar en mí, por su apoyo y por compartir conmigo, cada día, las dichas y sinsabores acaecidas a lo largo de esta tesis.

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) y proyecto «INREDIS: Interfaces de Relación entre el Entorno y las personas con Discapacidad» (proyecto CENIT).

Resumen

Los factores claves que deben tenerse en cuenta en el diseño de aplicaciones para redes de área personal inalámbricas son, fundamentalmente, la interoperatividad y el soporte de calidad de servicio, requisito imprescindible para garantizar la utilización eficiente del ancho de banda de las aplicaciones móviles basadas en este tipo de redes.

Esta tesis se ha centrado en el estudio del comportamiento de Bluetooth que, hoy por hoy, es la tecnología más extendida para la formación de redes de área personal y que, gracias a que su especificación incluye la definición de perfiles, garantiza la interoperatividad.

Sin embargo, hasta el momento, en el ámbito de las redes WPAN (*Wireless Personal Area Network*) Bluetooth, un número significativo de trabajos no contemplan el uso de perfiles, considerando que los datos se entregan directamente a niveles inferiores de la arquitectura Bluetooth y, obviando así, el *overhead* introducido por las capas superiores. Al respecto, es conveniente aclarar que esta opción no es factible en sistemas en los que se integran dispositivos reales con un determinado perfil, habitualmente SPP (*Serial Port Profile*).

Es precisamente el hecho de que una amplia variedad de dispositivos, que van desde periféricos de ordenadores personales hasta sensores médicos, implementen el perfil SPP para la transferencia de datos por Bluetooth lo que ha motivado, en esta tesis, el estudio de rendimiento, en términos de retardo extremo a extremo, para este perfil.

También se ha querido comparar el comportamiento del perfil SPP con el del perfil PAN (*Personal Area Network*), al que se le augura una futura expansión debido a que permite el encapsulamiento de tráfico IP sobre Bluetooth, aunque actualmente no esté soportado por ningún dispositivo comercial.

Concretamente, el objetivo de esta tesis ha sido estudiar el rendimiento de una WPAN Bluetooth, en términos del retardo extremo a extremo y del *throughput*, contemplando el uso de perfiles y evaluando el efecto de la configuración de la calidad de servicio y de las retransmisiones.

El estudio desarrollado ha consistido en formular expresiones matemáticas para calcular el retardo de transmisión mínimo en el sentido de transmisión maestro-esclavo, dependiendo del tipo de paquete utilizado y para un tamaño de datos de usuario arbitrario. Posteriormente, el modelo propuesto ha sido extendido para contemplar el *overhead* introducido por los niveles superiores, tanto en el caso del perfil SPP, como en el del perfil PAN. Adicionalmente, para el perfil SPP se ha completado el estudio analítico a fin de considerar la configuración de calidad, así como la ocurrencia de retransmisiones.

En una segunda fase, se ha llevado a cabo la validación del modelo mediante caracterización empírica, empleando dispositivos reales Bluetooth 1.1. A raíz de los resultados obtenidos se han introducido algunas correcciones que

permiten afinar el modelo, ajustándose mejor al comportamiento observado experimentalmente.

También se ha extendido el estudio de evaluación de la configuración de calidad al caso de una *piconet* Bluetooth de dos esclavos, proporcionando cotas mínima y máxima del retardo para cada sentido de transmisión, cotas, que tras verificarse empíricamente, han sido generalizadas y, de nuevo, validadas experimentalmente para una *piconet* Bluetooth de tres esclavos.

Finalmente, se han repetido las medidas realizadas para una *piconet* de un único esclavo utilizando dispositivos Bluetooth 1.2 y Bluetooth 2.0, pudiéndose concluir que las ecuaciones de retardo propuestas modelan aún mejor el comportamiento observado que en el caso de Bluetooth 1.1.

A modo de conclusión, cabe mencionar que en esta tesis se han establecido las condiciones que han de cumplirse para minimizar el tiempo de transmisión (y maximizar el *throughput*), teniendo en cuenta el *overhead* introducido por todos los protocolos de los niveles de la arquitectura Bluetooth asociados a un perfil. Además, se han definido ecuaciones del retardo y *throughput* para los perfiles SPP y PAN en una *piconet* de dos dispositivos y en el sentido de transmisión maestro-esclavo, analizando también para el perfil SPP el efecto de la configuración de calidad de servicio y de la ocurrencia de retransmisiones en los sentidos de comunicación maestro-esclavo y esclavo-maestro. Adicionalmente, se ha modelado, para el perfil SPP, el comportamiento de una *piconet* de varios dispositivos, proponiendo cotas del retardo de transmisión, y quedando, como línea futura, entre otras, para continuar investigando, la extensión del modelo para una *scatternet*.

Abstract

Interoperability and Quality of Service (QoS) support are the key issues to be considered in the scope of Wireless Personal Area Networks (WPAN). In fact, the QoS support is an essential requirement to guarantee an efficient employment of bandwidth for mobile applications based on WPANs. However, up to date, in the Bluetooth WPAN context, most published research does not pay attention to Bluetooth profiles and prefers to consider that user data are directly sent over low layers of the Bluetooth architecture. Consequently, the overhead introduced by high layers is neglected. This decision prevents extending most analytical results to actual Bluetooth devices, as they always integrate a specific profile, frequently SPP (*Serial Port Profile*).

Specifically, in this thesis, the fact that most medical commercial devices implement the SPP profile has led to study the performance for the SPP profile in terms of end-to-end delay and throughput.

Moreover, SPP has been compared to the PAN (*Personal Area Network*) profile behaviour, increasingly widespread although it is not currently supported by most Bluetooth devices.

The aim of this thesis is to study the performance of Bluetooth WPANs, in terms of end-to-end delay and throughput, considering the use of profiles and evaluating the impact of Quality of Service configuration and retransmissions on the transmission delay.

This thesis develops a model that calculates the minimal transmission delay in master-slave communications, depending on the type of packet used and the length of user data. Subsequently, the proposed model has been extended for SPP and PAN profiles in order to regard the overhead aggregated by every layer. In addition, for SPP profile, the analytical study has been completed to take into account the effect of the Quality of Service configuration and the occurrence of retransmissions.

In a second stage, an empirical validation of the model has been carried out using Bluetooth 1.1 devices. From the results obtained, several improvements have been introduced in order to tune the model and to track the real observed behaviour precisely.

Additionally, the study has been completed by evaluating Quality of Service configuration for a Bluetooth piconet with two slaves. Setting up lower and upper limits of delay for every direction of transmission has been proposed. These limits have been later verified empirically and generalised, and again validated for a Bluetooth piconet with three slaves.

Finally, the measurements performed for a Bluetooth 1.1 piconet with a single slave have been repeated with Bluetooth 1.2 and Bluetooth 2.0 devices, concluding that the delay equations defined in this thesis allow to model the behaviour even better than in the case of Bluetooth 1.1.

As a conclusion, this thesis specifies the conditions which have to be satisfied in order to minimise the end-to-end delay (and to maximise Bluetooth throughput), considering the overhead added by every layer of Bluetooth stack associated to a specific profile. Moreover, this thesis presents equations which have been defined to calculate the end-to-end delay and throughput for SPP and PAN profiles, in a Bluetooth piconet with two devices and for master-slave direction of transmission. For the case of SPP profile, these equations have been extended to consider the effect of the configuration of QoS parameters and the occurrence of retransmissions. In addition, for SPP profile and for the case of a piconet with several slaves, the delay has been modeled by means of the definition of lower and upper limits, being a future line of research the study of the scatternet behaviour.

Índice general

Índice de figuras	v
Índice de tablas	xvii
Lista de acrónimos, abreviaturas y símbolos	xix
1. Introducción	1
2. Fundamentos Teóricos de Bluetooth	9
2.1. Bluetooth 1.1	11
2.1.1. Nivel RF (<i>Radio Frequency</i>)	13
2.1.2. Nivel BB (<i>Baseband</i>)	15
2.1.2.1. Canal Físico	15
2.1.2.2. Enlaces físicos	16
2.1.2.3. Estados y modos de operación	18
2.1.2.4. Estructura y tipos de los paquetes BB	22
2.1.2.5. Corrección de Errores	28
2.1.3. Nivel LM (<i>Link Manager</i>)	30
2.1.4. Nivel HCI (<i>Host Controller Interface</i>)	31
2.1.5. Nivel L2CAP (<i>Logical Link Control and Adaptation Protocol</i>)	31

2.2. Bluetooth 1.2	34
2.3. Bluetooth 2.0 + EDR	35
2.4. Perfiles Bluetooth	36
2.4.1. Perfil SPP	38
2.4.1.1. Protocolo RFCOMM	39
2.4.2. Perfil PAN	40
2.4.2.1. Protocolo BNEP (<i>Bluetooth Network Encapsulation Protocol</i>)	41
3. Revisión Bibliográfica:	
Calidad de Servicio en Redes de Área Personal Bluetooth	43
3.1. Efecto de las interferencias en la calidad de servicio	44
3.1.1. Coexistencia Bluetooth - IEEE 802.11	44
3.1.2. Coexistencia entre <i>piconets</i>	47
3.2. Calidad de servicio en <i>piconets</i>	50
4. Estudio teórico del rendimiento en <i>piconets</i> Bluetooth utilizando perfiles	59
4.1. Perfiles SPP y PAN: Efecto del <i>overhead</i> y del <i>segmentation mismatch</i> en el retardo	61
4.1.1. Introducción	61
4.1.2. Cálculo del retardo mínimo de transmisión	62
4.1.3. Cálculo del retardo para el perfil SPP	67
4.1.4. Cálculo del retardo para el perfil PAN	71
4.1.5. Comparación del retardo mínimo con el de los perfiles SPP y PAN	76

4.1.6.	Estimación del <i>throughput</i>	79
4.1.7.	Discusión	86
4.2.	Perfil SPP: Efecto de la configuración de calidad de servicio en el retardo de transmisión	89
4.3.	Perfil SPP: Efecto de las retransmisiones en el retardo de transmisión	96
5.	Evaluación de los modelos mediante caracterización empírica	109
5.1.	Metodología de medida	111
5.2.	Comparación del retardo teórico y medido: Perfiles SPP y PAN .	115
5.3.	Efecto de la configuración de la calidad de servicio	121
5.3.1.	Medidas para una <i>piconet</i> de dos dispositivos	123
5.3.1.1.	Sentido de transmisión maestro-esclavo	124
5.3.1.2.	Sentido de transmisión esclavo-maestro	128
5.3.1.3.	Comparación de los retardos de transmisión en los sentidos maestro-esclavo y esclavo-maestro .	131
5.3.1.4.	Corrección del retardo en la transmisión maestro-esclavo	138
5.3.1.5.	Corrección del retardo en la transmisión esclavo-maestro	155
5.3.2.	Medidas para una <i>piconet</i> de tres dispositivos	164
5.3.2.1.	Modelo teórico para una <i>piconet</i> de dos esclavos .	165
5.3.2.2.	Validación del modelo teórico para una <i>piconet</i> de dos esclavos	170
5.3.3.	Generalización del modelo para <i>piconets</i> de más de dos dispositivos	186

5.3.4. Medidas para una <i>piconet</i> de cuatro dispositivos	190
5.4. Efecto de las retransmisiones	202
6. Conclusiones y líneas futuras	213
6.1. Conclusiones	215
6.1.1. Conclusiones sobre el estudio comparativo de los perfiles SPP y PAN	216
6.1.2. Conclusiones sobre el estudio del efecto de la configura- ción de calidad realizado para el perfil SPP	219
6.1.3. Conclusiones sobre el estudio del efecto de las retransmi- siones realizado para el perfil SPP	222
6.2. Líneas futuras	223
A. Medidas realizadas con Bluetooth 1.2 y Bluetooth 2.0	227
Bibliografía	239

Índice de figuras

2.1. Arquitectura de protocolos Bluetooth 1.1	13
2.2. Correspondencia entre la arquitecturas Bluetooth, IEEE 802 y modelo OSI (fuente: estándar 802.15.1-2002 [IEEE 802 02])	14
2.3. Esquema de transmisión de paquetes <i>multi-slot</i> (fuente: parte B de la especificación [Bluetooth SIG01c])	17
2.4. Diagrama de estados definido en Bluetooth 1.1 (fuente: parte B de la especificación [Bluetooth SIG01c])	19
2.5. Estructura de los paquetes Bluetooth y formato del campo de datos (el tamaño de los campos está indicado en bits)	25
4.1. Arquitectura de protocolos Bluetooth	63
4.2. Estructura de la trama del Modo de Operación Básico del protocolo TS 07.10 (fuente: parte F:1 de la especificación [Bluetooth SIG01c])	68
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 4.7	70

4.4. PDU de L2CAP en Bluetooth 1.1 (fuente: parte D de la especificación [Bluetooth SIG01c]) y Bluetooth 1.2 (fuente: parte A del volumen 3 de la especificación [Bluetooth SIG03a])	70
4.5. Formatos de las cabeceras de los paquetes BNEP [Bluetooth SIG03b]	72
4.6. Fragmentación realizada en el nivel IP para que, al añadirle la cabecera IP, no se exceda la MTU M'_B	74
4.7. Comparación del retardo mínimo ($t_{TX}(N)$) con el correspondiente a los perfiles PAN ($t_U(N)$) y SPP ($t_R(N)$)	77
4.8. Comparación del <i>throughput</i> máximo con el correspondiente a los perfiles PAN y SPP	84
4.9. Efecto del <i>segmentation mismatch</i> sobre el <i>throughput</i> , para paquetes DH cuando se emplea el perfil SPP	85
4.10. Comparación del retardo para los perfiles PAN y SPP, con paquetes DM, fijando M_R y $N1$, a 4096 y 4091 octetos, respectivamente, satisfaciéndose así las condiciones $N1 \geq N$ y $M_R \geq N1 + 5$	88
4.11. Efecto del intervalo de <i>poll</i> en la transmisión de fragmentos consecutivos para el sentido de comunicación maestro-esclavo . .	91
4.12. Efecto del intervalo de <i>poll</i> en la transmisión de fragmentos consecutivos para el sentido de comunicación esclavo-maestro . .	91
4.13. Retardo máximo de transmisión en el sentido maestro-esclavo para el perfil SPP, teniendo en cuenta el efecto del intervalo de <i>poll</i>	94
4.14. Retardo máximo de transmisión en el sentido esclavo-maestro para el perfil SPP, teniendo en cuenta el efecto del intervalo de <i>poll</i>	95

4.15. Tiempo transcurrido desde que se inicia la transmisión de un paquete de 5 <i>slots</i> hasta que se realiza la retransmisión, particularizando $t_{RTx}(N)$ al valor máximo, $t_{RTx_{max}}(N)$	104
4.16. Tiempo transcurrido desde que se inicia la transmisión de un paquete de 5 <i>slots</i> hasta que se realiza la retransmisión, particularizando $t_{RTx}(N)$ al valor mínimo, $t_{RTx_{min}}(N)$	104
4.17. Efecto de la tasa BER en las transmisiones maestro-esclavo para el perfil SPP	107
4.18. Efecto de la tasa BER en las transmisiones esclavo-maestro para el perfil SPP	108
5.1. Esquema general de pruebas empleado para la caracterización del retardo de transmisión Bluetooth, utilizado para evitar las interferencias	111
5.2. Comparación de los retardos de transmisión maestro-esclavo obtenidos a nivel RFCOMM y a nivel HCI, utilizando paquetes DH	114
5.3. Comparación del retardo medio de transmisión medido con el calculado analíticamente, mediante la ecuación 4.6, para el perfil SPP	117
5.4. Comparación del retardo medio de transmisión medido con el calculado analíticamente, mediante la ecuación 4.9, para el perfil PAN	118
5.5. Comparación del retardo medio de transmisión maestro-esclavo para valores de T_{poll} (asimilable al parámetro <i>Latency</i>) inferiores a 15 ms	125

5.6. Comparación del retardo medio de transmisión maestro-esclavo para distintos valores del parámetro T_{poll}	126
5.7. Comparación del retardo medio de transmisión esclavo-maestro para valores del parámetro T_{poll} inferiores a 15 ms	129
5.8. Comparación del retardo medio de transmisión esclavo-maestro para distintos valores del parámetro T_{poll}	130
5.9. Comparación de los retardos medios de transmisión en los sentidos maestro-esclavo y esclavo-maestro para un valor del parámetro T_{poll} de 5 ms	132
5.10. Comparación de los retardos medios de transmisión en los sentidos maestro-esclavo y esclavo-maestro para un valor del parámetro T_{poll} de 10 ms	133
5.11. Comparación de los retardos medios de transmisión en los sentidos maestro-esclavo y esclavo-maestro para un valor del parámetro T_{poll} de 15 ms	134
5.12. Comparación de los retardos medios de transmisión en los sentidos maestro-esclavo y esclavo-maestro para un valor del parámetro T_{poll} de 20 ms	135
5.13. Comparación de los retardos medios de transmisión en los sentidos maestro-esclavo y esclavo-maestro para un valor del parámetro T_{poll} de 25 ms	136
5.14. Comparación de los retardos de transmisión maestro-esclavo medido y teórico corregido para un valor del parámetro T_{poll} de 5 ms	140

5.15. Comparación de los retardos de transmisión maestro-esclavo medido y teórico corregido para un valor del parámetro T_{poll} de 10 ms	141
5.16. Comparación de los retardos de transmisión maestro-esclavo medido y teórico corregido para un valor del parámetro T_{poll} de 15 ms	142
5.17. Comparación de los retardos de transmisión maestro-esclavo medido y teórico corregido para un valor del parámetro T_{poll} de 20 ms	143
5.18. Comparación de los retardos de transmisión maestro-esclavo medido y teórico corregido para un valor del parámetro T_{poll} de 25 ms	144
5.19. Comparación de los retardos de transmisión maestro-esclavo medido y teórico, corregido y mejorado, para un valor del parámetro T_{poll} de 5 ms	146
5.20. Comparación de los retardos de transmisión maestro-esclavo medido y teórico, corregido y mejorado, para un valor del parámetro T_{poll} de 10 ms	147
5.21. Comparación de los retardos de transmisión maestro-esclavo medido y teórico, corregido y mejorado, para un valor del parámetro T_{poll} de 15 ms	148
5.22. Comparación de los retardos de transmisión maestro-esclavo medido y teórico, corregido y mejorado, para un valor del parámetro T_{poll} de 20 ms	149

5.23. Comparación de los retardos de transmisión maestro-esclavo medido y teórico, corregido y mejorado, para un valor del parámetro T_{poll} de 25 ms	150
5.24. Comparación de los retardos de transmisión maestro-esclavo medido y teórico, corregido y mejorado con un desplazamiento de -1 ms , para un valor del parámetro T_{poll} de 15 ms	152
5.25. Comparación de los retardos de transmisión maestro-esclavo medido y teórico, corregido y mejorado con un desplazamiento de -1 ms , para un valor del parámetro T_{poll} de 20 ms	153
5.26. Comparación de los retardos de transmisión maestro-esclavo medido y teórico, corregido y mejorado con un desplazamiento de -1 ms , para un valor del parámetro T_{poll} de 25 ms	154
5.27. 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	156
5.28. Comparación de los retardos de transmisión esclavo-maestro medidos y teórico corregido para un valor del parámetro T_{poll} de 5 ms	159
5.29. Comparación de los retardos de transmisión esclavo-maestro medidos y teórico corregido para un valor del parámetro T_{poll} de 10 ms	160
5.30. Comparación de los retardos de transmisión esclavo-maestro medidos y teórico corregido para un valor del parámetro T_{poll} de 15 ms	161

5.31. Comparación de los retardos de transmisión esclavo-maestro medidos y teórico corregido para un valor del parámetro T_{poll} de 20 ms	162
5.32. Comparación de los retardos de transmisión esclavo-maestro medidos y teórico corregido con paquetes DM para un valor del parámetro T_{poll} de 25 ms	163
5.33. Tiempo mínimo entre la transmisión de fragmentos consecutivos a un mismo esclavo en presencia de un segundo esclavo para el que no hay datos pendientes de transmisión	167
5.34. Tiempo máximo entre la transmisión de fragmentos consecutivos a un mismo esclavo, debido a la alternancia de las transmisiones a los esclavos E1 y E2 de fragmentos intermedios en paquetes de 5 <i>slots</i>	167
5.35. Tiempo mínimo entre la transmisión de fragmentos consecutivos de un mismo esclavo en presencia de un segundo esclavo que tiene datos pendientes de transmisión	168
5.36. Tiempo máximo entre las transmisiones de fragmentos consecutivos de un mismo esclavo, debido a la alternancia de las transmisiones de los esclavos E1 y E2 de fragmentos intermedios en paquetes de 5 <i>slots</i>	168
5.37. Escenarios de prueba en una <i>piconet</i> de 3 dispositivos	171
5.38. Escenario 1: Cotas mínimas y máximas, dadas por las ecuaciones 5.9 y 5.10 (subfigura (a)), y 5.12 y 5.13 (subfigura (b)), y retardo de transmisión medido en cada conexión, con paquetes DH y un valor del parámetro T_{poll} de 5 ms	175

5.39. Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.9 y 5.10, y retardo de transmisión medido en cada conexión, con paquetes DH y un valor del parámetro T_{poll} de 5 ms	176
5.40. Escenario 3: Cotas mínima y máxima, dadas por las ecuaciones 5.12 y 5.13, y retardo de transmisión medido en cada conexión, con paquetes DH y un valor del parámetro T_{poll} de 5 ms	177
5.41. Escenario 1: Cotas mínimas y máximas, dadas por las ecuaciones 5.9 y 5.10 (subfigura (a)), y 5.12 y 5.13 (subfigura (b)), y retardo de transmisión medido en cada conexión, con paquetes DH y un valor del parámetro T_{poll} de 10 ms	178
5.42. Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.9 y 5.10, y retardo de transmisión medido en cada conexión, con paquetes DH y un valor del parámetro T_{poll} de 10 ms	179
5.43. Escenario 3: Cotas mínima y máxima, dadas por las ecuaciones 5.12 y 5.13, y retardo de transmisión medido en cada conexión, con paquetes DH y un valor del parámetro T_{poll} de 10 ms	180
5.44. Escenario 1: Cotas mínimas y máximas, dadas por las ecuaciones 5.9 y 5.10 (subfigura (a)), y 5.12 y 5.13 (subfigura (b)), y retardo de transmisión medido en cada conexión, con paquetes DH y un valor del parámetro T_{poll} de 15 ms	181
5.45. Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.9 y 5.10, y retardo de transmisión medido en cada conexión, con paquetes DH y un valor del parámetro T_{poll} de 15 ms	182

5.46. Escenario 3: Cotas mínima y máxima, dadas por las ecuaciones 5.12 y 5.13, y retardo de transmisión medido en cada conexión, con paquetes DH y un valor del parámetro T_{poll} de 15 ms	183
5.47. Escenarios 1 y 2: Cotas mínima y máxima, dadas por las ecuaciones 5.9 y 5.10 y retardo de transmisión medido en la conexión maestro-esclavo, con paquetes DH y un valor del parámetro T_{poll} de 20 ms	184
5.48. Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.9 y 5.10, y retardo de transmisión medido en cada conexión, con paquetes DH, un valor del parámetro T_{poll} de 5 ms y periodos distintos	185
5.49. Cotas mínima y máxima, dadas por las ecuaciones 5.17 y 5.18, y retardo medido en las transmisiones del maestro a los esclavos E1 y E2, con paquetes DH y un valor del parámetro T_{poll} de 5 ms . . .	192
5.50. Escenario 1: Cotas mínima y máxima, dadas por las ecuaciones 5.17 y 5.18, y retardo medido en la transmisión del maestro al esclavo E3, con paquetes DH y un valor del parámetro T_{poll} de 5 ms	193
5.51. Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.20 y 5.21, y retardo medido en la transmisión del esclavo E1 al maestro, con paquetes DH y un valor del parámetro T_{poll} de 5 ms	193
5.52. Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.20 y 5.21, y retardo medido en las transmisiones de los esclavos E1 y E2 al maestro, con paquetes DH y un valor del parámetro T_{poll} de 5 ms	194

5.53. Cotas mínima y máxima, dadas por las ecuaciones 5.17 y 5.18, y retardo medido en las transmisiones del maestro a los esclavos E1 y E2, con paquetes DH y un valor del parámetro T_{poll} de 10 ms . . .	195
5.54. Escenario 1: Cotas mínima y máxima, dadas por las ecuaciones 5.17 y 5.18, y retardo medido en la transmisión del maestro al esclavo E3, con paquetes DH y un valor del parámetro T_{poll} de 10 ms	196
5.55. Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.20 y 5.21, y retardo medido en la transmisión del esclavo E1 al maestro, con paquetes DH y un valor del parámetro T_{poll} de 10 ms	196
5.56. Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.20 y 5.21, y retardo medido en las transmisiones de los esclavos E1 y E2 al maestro, con paquetes DH y un valor del parámetro T_{poll} de 10 ms	197
5.57. Escenario 1: Cotas mínima y máxima, dadas por las ecuaciones 5.17 y 5.18, y retardo medido en las transmisiones del maestro a los esclavos E1 y E2, con paquetes DH y un valor del parámetro T_{poll} de 15 ms	198
5.58. Escenario 1: Cotas mínima y máxima, dadas por las ecuaciones 5.17 y 5.18, y retardo medido en la transmisión del maestro al esclavo E3, con paquetes DH y un valor del parámetro T_{poll} de 15 ms	199
5.59. Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.20 y 5.21, y retardo medido en la transmisión del esclavo E1 al maestro, con paquetes DH y un valor del parámetro T_{poll} de 15 ms	199

5.60. Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.20 y 5.21, y retardo medido en las transmisiones de los esclavos E1 y E2 al maestro, con paquetes DH y un valor del parámetro T_{poll} de 15 ms	200
5.61. Cotas mínima y máxima, dadas por las ecuaciones 5.17 y 5.18, y retardo medido en las transmisiones del maestro a un esclavo, con paquetes DH (cuando los otros dos esclavos transmiten al maestro)	201
5.62. Representación de la ecuación 5.22, que define la relación entre la tasa BER y el parámetro LQ, según especifica el fabricante CSR [CSR]	204
5.63. Esquema de pruebas utilizado en la evaluación del impacto de las retransmisiones sobre el retardo en transmisiones Bluetooth, utilizando el perfil SPP	204
5.64. Transmisión esclavo-maestro con pérdidas para un valor del parámetro T_{poll} de 10 ms. BER media estimada: $3,808 \times 10^{-3}$. . .	207
5.65. Transmisión esclavo-maestro con pérdidas para un valor del parámetro T_{poll} de 5 ms. BER media estimada: $9,357 \times 10^{-3}$	208
5.66. Transmisión esclavo-maestro con pérdidas para un valor del parámetro T_{poll} de 5 ms. BER media estimada: $1,033 \times 10^{-2}$	209
5.67. Transmisión maestro-esclavo con pérdidas para un valor del parámetro T_{poll} de 5 ms. BER media estimada: $3,506 \times 10^{-3}$	210
5.68. Transmisión maestro-esclavo con retransmisiones para un valor del parámetro T_{poll} de 10 ms. BER media estimada: $4,856 \times 10^{-3}$.	211
5.69. Transmisión maestro-esclavo con pérdidas para un valor del parámetro T_{poll} de 10 ms. BER media estimada = $6,85 \times 10^{-3}$. . .	212

A.1. Comparación de los retardos de transmisión obtenidos utilizando dispositivos Bluetooth 1.1 y Bluetooth 2.0 para diferentes valores del parámetro T_{poll}	229
A.2. Curva del retardo de transmisión obtenida en el modo EDR de Bluetooth 2.0 para un valor del parámetro T_{poll} de 25 ms, en la que se han marcado, con líneas verticales, las longitudes 367 octetos, 552 octetos, 679 octetos y 1021 octetos.	231
A.3. Comparación de los retardos de transmisión obtenidos utilizando dispositivos Bluetooth 1.1 y Bluetooth 2.0 para diferentes valores del parámetro T_{poll}	234
A.4. Comparación de los retardos de transmisión maestro-esclavo medido y teórico para valores del parámetro T_{poll} de 5 ms, 10 ms, 15 ms, 20 ms y 25 ms	235
A.5. Comparación de los retardos de transmisión esclavo-maestro medido y teórico para valores del parámetro T_{poll} de 5 ms	235
A.6. Comparación de los retardos de transmisión esclavo-maestro medido y teórico para valores del parámetro T_{poll} de 10 ms y 15 ms	236
A.7. Comparación de los retardos de transmisión esclavo-maestro medido y teórico para valores del parámetro T_{poll} de 20 ms y 25 ms	237

Índice de tablas

2.1. Clases de dispositivos Bluetooth	15
2.2. Tipos de paquetes Bluetooth	23
2.3. Tipos de paquetes ACL	27
2.4. Tipos de tramas TS 07.10 utilizadas por RFCOMM	40
4.1. <i>Throughput</i> máximo para los distintos tipos de paquetes ACL . . .	80
5.1. Paquetes Bluetooth necesarios para la transmisión de 1010 octetos de datos de usuario utilizando el perfil SPP	119
5.2. Paquetes Bluetooth necesarios para la transmisión de 1480 octetos de datos de usuario utilizando el perfil PAN	120
5.3. Valores del parámetro <i>Latency</i> (y del intervalo de <i>poll</i> al que equivale) para los que se han realizado las medidas de evaluación del efecto de la configuración de la calidad de servicio	123
A.1. Tipos de paquetes ACL definidos en Bluetooth 2.0 + EDR	230

Lista de acrónimos, abreviaturas y símbolos

Acrónimos y abreviaturas

ACK Acknowledge.

ACL (Bluetooth 1.1) Asynchronous Connection-Less.

AFH Adaptive Frequency Hopping.

AM_ADDR Active Member Address.

AR_ADDR Access Request Address.

ARQ Automatic Repeat reQuest.

ARQN Automatic Repeat reQuest Number.

ATM Asynchronous Transfer Mode.

BAN Body Area Network.

BB Baseband.

BCS	Bluetooth Carrier Sense.
BD_ADDR	Bluetooth Address.
BER	Bit Error Rate.
BNEP	Bluetooth Network Encapsulation Protocol.
CAC	Channel Access Code.
CRC	Cyclic Redundancy Check.
CSR	Cambridge Silicon Radio Plc..
DAC	Device Access Code.
DCE	Data Circuit-Terminating Equipment.
DCT	Dual Channel Transmission.
DH	Data High-Rate.
DISC	Disconnect.
DLC	Data Link Connection.
DM	Data Medium-Rate.
DM (RFCOMM)	Disconnected Mode.
DPSK	Differential PSK.
DQPSK	Differential Quaternary PSK.
DTE	Data Terminal Equipment.

DUN	Dial-Up Networking Profile.
DV	Data-Voice.
EDR	Enhanced Data Rate.
EIA	Electronic Industries Alliance.
FEC	Forward Error Correction.
GAP	Generic Access Profile.
GFSK	Gaussian Frequency Shift Keying.
HCI	Host Controller Interface.
HCTL	Host Controller Transport Layer.
HEC	Header Error Check.
HID	Human Interface Device.
HV	High-quality Voice.
IAC	Inquiry Access Code.
IEEE	Institute of Electronic and Electrical Engineering.
IP	Internet Protocol.
ISM	Industrial, Scientific and Medical.
L2CAP	Logical Link Control and Adaptation Protocol.
L_CH	Logical Channel.

LAN	Local Area Network.
LM	Link Manager.
LMP	Link Management Protocol.
LQ	Link Quality.
MAC	Medium Access Control.
MDP	Medical Device Profile.
MSS	Maximum Segment Size.
MTU	Maximum Transfer Unit.
N1	Maximum Frame Size.
NAK	Negative Acknowledge.
NAP	Network Access Point.
OSI	Open Systems Interconnection.
PAN	Personal Area Network.
PANU	Personal Area Network User.
PDA	Personal Digital Assistant.
PDU	Protocol Data Unit.
PER	Packet Error Rate.
PHY	Physical.

PM_ADDR	Parked Member Address.
PPP	Point-to-Point Protocol.
PSK	Phase Shift Keying.
QoS	Quality of Service.
RAM	Random Access Memory.
RF	Radio Frequency.
RFCOMM	Serial Cable Emulation Protocol basado en la norma ETSI TS 07.10.
RR	Round Robin.
SABM	Set Asynchronous Balanced Mode.
SAR	Segmentation And Reassembling.
SCO	Synchronous Connection-Oriented.
SCT	Single Channel Transmission.
SDAP	Service Discovery Application Profile.
SDP	Service Discovery Protocol.
SIG	Special Interest Group.
SNR	Signal to Noise Ratio.
SPP	Serial Port Profile.
TCP	Transmission Control Protocol,.

TCS	Telephony Control- Protocol Specification.
TDD	Time Division Duplex.
TIA	Telecommunications Industry Association.
UA	Unnumbered Acknowledgement.
UIH	Unnumbered information with header check.
WPAN	Wireless Personal Area Networks.

Símbolos

$\delta(N)$	<i>Offset</i> que se incluye en el cálculo del retardo de transmisión para SPP en el sentido esclavo-maestro, y que afecta a longitudes datos que exceden la capacidad de un paquete de 5 <i>slots</i> .
Δ_{poll}	Corrección introducida en la cota superior del retardo de transmisión para el perfil SPP $t_{RM_{max}}(N)$ para tener en cuenta que el intervalo de <i>poll</i> configurado pueda o no garantizarse en una <i>piconet</i> de tres dispositivos.
$\Delta_{poll}(n_e)$	Corrección introducida en la cota superior del retardo de transmisión para el perfil SPP $t_{RM_{max}}(N, n_e)$ para tener en cuenta que el intervalo de <i>poll</i> configurado pueda o no garantizarse en una <i>piconet</i> de n_e esclavos.
$\lceil Y \rceil$	Operador que redondea el valor de Y al entero superior.
$\lfloor Y \rfloor$	Operador que redondea el valor de Y al entero inferior.
$\overline{N_{RTx}}$	Número medio de retransmisiones expresado en función de la tasa <i>PER</i> , también notada como p .
$\overline{t_{ACK_{per}}}(N)$	Promedio de $t_{ACK}(N)$ teniendo en cuenta el efecto de las retransmisiones.
$\overline{t_{R_{Emin}}}(N)$	Cota mínima del retardo de transmisión para el perfil SPP y una <i>piconet</i> de tres dispositivos en el sentido de transmisión esclavo-maestro.

$\overline{t_{R_{E_{min}}}}(N, n_e)$	Cota mínima del retardo de transmisión para el perfil SPP y una <i>piconet</i> de n_e esclavos en el sentido de transmisión esclavo-maestro.
$\overline{t_{R_{E_{per}}}}(N)$	Cota máxima del retardo para el perfil SPP en el sentido de transmisión esclavo-maestro, teniendo en cuenta las retransmisiones.
$\overline{t_{R_E}}(N)$	Retardo medio de transmisión, para SPP, en el sentido esclavo-maestro, teniendo en cuenta la configuración de calidad.
$\overline{t_{R_{M_{min}}}}(N)$	Cota mínima del retardo de transmisión para el perfil SPP y una <i>piconet</i> de tres dispositivos en el sentido de transmisión maestro-esclavo.
$\overline{t_{R_{M_{min}}}}(N, n_e)$	Cota mínima del retardo de transmisión para el perfil SPP y una <i>piconet</i> de n_e esclavos en el sentido de transmisión maestro-esclavo.
$\overline{t_{R_{M_{per}}}}(N)$	Cota máxima del retardo 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 SPP, en el sentido maestro-esclavo, teniendo en cuenta la configuración de calidad.
$\overline{t_{TX_{per}}}(N)$	Promedio de $t_{TX}(N)$ teniendo en cuenta el efecto de las retransmisiones.
BER	Tasa de bits erróneos.
C_{BB}	Tamaño del <i>overhead</i> 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.
C_B	Tamaño de la cabecera BNEP.

$C_{df}(N)$	Tamaño de la cabecera del <i>payload</i> .
C_I	Tamaño de la cabecera IP.
C_L	Tamaño de la cabecera de la PDU de L2CAP en Bluetooth 1.1 equivalente a la PDU de L2CAP del modo básico del protocolo L2CAP en Bluetooth v1.2: 4 octetos.
C_U	Tamaño de la cabecera UDP: 8 octetos.
d^*	Distancia de Hamming más pequeña entre las distintas palabras de código, siendo la distancia de Hamming entre dos secuencias, el número de bits en que se diferencian.
k	Número de símbolos de entrada, o símbolos de información en un código de Hamming (n,k) .
L_1	Longitud máxima del <i>payload</i> de un paquete Bluetooth de 1 <i>slot</i> .
L_3	Longitud máxima del <i>payload</i> de un paquete Bluetooth de 3 <i>slots</i> .
L_5	Longitud máxima del <i>payload</i> de un paquete Bluetooth de 5 <i>slots</i> .
L_{ffi}	Longitud del fragmento final obtenido tras la fragmentación realizada a nivel IP para no exceder la MTU M'_B .
L_{ffR}	Longitud del fragmento final resultante de dividir los N octetos de datos en fragmentos de L_R octetos.
L_R	Longitud a la que RFCOMM tendrá que 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 $N1$.
m	Número de símbolos de paridad en un código de Hamming (n,k) .
M'_B	MTU de BNEP.
M_B	MTU de L2CAP para BNEP.
M_R	MTU de L2CAP para RFCOMM.
N	Tamaño de los datos de usuario en octetos.
n	Número de símbolos de salida, es decir, la longitud de la palabra codificada, en un código de Hamming (n,k) .
$n_{b_u}(N)$	Número de bits de información útiles a transmitir para N octetos de usuario, es decir, excluyendo la información de redundancia, FEC y CRC, así como la cabecera del payload.
$n_b(N)$	Número total de bits transmitidos para N octetos de usuario, obtenido como la suma de $n_d(N)$ y C_{BB} .
$n_d(N)$	Número de bits del campo de datos (<i>Data Field</i>), 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.
n_e	Número de esclavos participantes, en modo activo, en la piconet.
n_{fnf_I}	Número de fragmentos no finales en los que IP tiene que fragmentar el datagrama UDP para que al añadirle la cabecera IP no se exceda la MTU M'_B .

n_{fnf_R}	Número de fragmentos RFCOMM no finales, teniendo en cuenta que los N octetos de datos, se dividen en fragmentos de L_R octetos.
$n_g(N)$	Número de grupos de 15 bits de los que consta un paquete DM.
n_{poll}	Número de <i>slots</i> correspondientes al <i>Poll interval</i> , T_{poll} .
n_{su}	Número de <i>slots</i> ocupados por el paquete que se utiliza en la transmisión de los $n_{bu}(N)$ de información.
$n_{slots}(N)$	Número de <i>slots</i> que supone la transmisión de los $n_b(N)$ bits más el <i>overhead</i> . Es utilizado para el cálculo del <i>throughput</i> real, $Thr(N)$.
$O_R(L)$	Tamaño del <i>overhead</i> introducido por el protocolo RFCOMM: 4 ó 5 octetos.
p	Tasa de paquetes erróneos PER(N).
P_b	Probabilidad de que un bloque de 15 bits de un paquete DM se pierda porque contenga errores no recuperables.
$PER(N)$	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.
$PER_c(N)$	Tasa PER para paquetes DH y DM expresada en función de la tasa BER y teniendo en cuenta que los errores se produzcan en la cabecera del paquete.
$PER_{DH_d}(N)$	Tasa PER para paquetes DH expresada en función de la tasa BER y teniendo en cuenta que los errores se produzcan en el campo de datos.

- $PER_{DM_d}(N)$ Tasa PER para paquetes DM expresada en función de la tasa BER, calculada teniendo en cuenta que los errores se produzcan en el campo de datos.
- t Capacidad de corrección de errores aleatorios en un código de Hamming (n,k) .
- $t_{ACK_{max}}(N, n_e)$ Cota máxima de la componente $t_{ACK}(N)$ para una *piconet* de n_e dispositivos.
- $t_{ACK_{max}}(N)$ Cota máxima de la componente $t_{ACK}(N)$ para una *piconet* de tres dispositivos.
- $t_{ACK_{min}}(N)$ Cota mínima de la componente $t_{ACK}(N)$ para una *piconet* de tres dispositivos.
- $t_{ACK_{min}}(N)$ Cota mínima de la componente $t_{ACK}(N)$ para una *piconet* de tres dispositivos.
- $t_{ACK_{poll}}(N)$ Redefinición de $t_{ACK}(N)$ teniendo en cuenta la configuración de calidad para los sentidos de transmisión maestro- esclavo y esclavo- maestro.
- $t_{ACK}(N)$ Tiempo empleado en transmitir y confirmar cada uno de los paquetes del nivel BB necesarios para enviar N octetos de datos.
- t_b Tiempo de transmisión de un bit: $1 \mu s$.
- $t_I(N)$ Retardo de transmisión de N octetos sobre IP utilizando el protocolo BNEP teniendo en cuenta el *overhead* introducido por UDP, IP,

BNEP y L2CAP, así como la fragmentación realizada por cada nivel para no exceder la MTU del nivel subyacente.

- T_{poll} *Poll interval* expresado en microsegundos.
- $t_{R_{E_{max}}}(N)$ Cota máxima del retardo de transmisión para el perfil SPP y una *piconet* de tres dispositivos en el sentido de transmisión esclavo-maestro.
- $t_{R_{E_{min}}}(N, n_e)$ Cota máxima del retardo de transmisión para el perfil SPP y una *piconet* de n_e esclavos en el sentido de transmisión esclavo-maestro.
- $t_{R_E}(N)$ Cota máxima del retardo de transmisión, para SPP, en el sentido esclavo-maestro, teniendo en cuenta la configuración de calidad .
- $t_{R_{M_{max}}}(N, n_e)$ Cota máxima del retardo de transmisión para el perfil SPP y una *piconet* de n_e esclavos en el sentido de transmisión maestro-esclavo.
- $t_{R_{M_{max}}}(N)$ Cota máxima del retardo de transmisión para el perfil SPP y una *piconet* de tres dispositivos en el sentido de transmisión maestro-esclavo.
- $t_{R_M}(N)$ Cota máxima del retardo de transmisión, para SPP, en el sentido maestro-esclavo, teniendo en cuenta la configuración de calidad.
- $t_{RTx_{max}}(N)$ Cota máxima de $t_{RTx}(N)$.
- $t_{RTx_{min}}(N)$ Cota mínima de $t_{RTx}(N)$.
- $t_{RTx}(N)$ Tiempo transcurrido desde que se inicia la transmisión de un paquete hasta que se realiza la retransmisión.

$t_R(N)$	Tiempo de transmisión de N octetos mediante RFCOMM teniendo en cuenta el <i>overhead</i> introducido por RFCOMM y L2CAP, así como la fragmentación realizada por RFCOMM para no exceder la MTU de L2CAP (M_R).
T_S	Duración de un <i>slot</i> : $625 \mu s$.
$t_{TX_{poll}}(N)$	Redefinición de $t_{TX}(N)$ teniendo en cuenta la configuración de calidad para los sentidos de transmisión maestro- esclavo y esclavo- maestro.
$t_{TX}(N)$	Tiempo empleado en la transmisión de N 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.
$t_U(N)$	Retardo de transmisión de N octetos sobre UDP, calculado en función de $t_I(N)$.
$Thr(N)$	<i>Throughput</i> real, calculado teniendo en cuenta el <i>overhead</i> añadido a los bits de información por cada uno de los niveles.
$Thr_{MAX}(N)$	<i>Throughput</i> máximo teórico alcanzable en la transmisión con un determinado paquete Bluetooth, asumiendo que no se realiza ninguna retransmisión y sin tener en cuenta el <i>overhead</i> añadido a los bits transmitidos.

Capítulo 1

Introducción

Las redes WPAN (*Wireless Personal Area Networks*) son redes de área personal inalámbricas de corto o muy corto alcance (desde unos pocos centímetros hasta unos 10 metros) concebidas para intercambiar información entre los dispositivos existentes al alcance de una persona.

Las redes WPAN pueden utilizarse tanto para sustituir los cables entre equipos y periféricos por conexiones de corto alcance temporales o permanentes, como para posibilitar la computación ubicua (*pervasive computing*) o el despliegue de redes *ad hoc* [M. Cordeiro04, Shinde05, Záruba04].

El comité IEEE 802 [IEEE 802], al percatarse de la importancia de las redes inalámbricas de corto alcance, creó el grupo de trabajo 802.15 con la finalidad de estandarizar protocolos e interfaces para WPAN.

Concretamente, se definieron tres clases de WPAN, diferenciadas por la tasa de transferencia, el consumo de batería y la calidad de servicio (QoS, *Quality of*

Service): (a) WPAN 802.15.3 (*High Rate WPAN*) [IEEE 802 03a] adecuadas para aplicaciones multimedia con requisitos de QoS muy restrictivos, (b) WPAN 802.15.1 (*Medium Rate WPAN*) [IEEE 802 02] orientadas a la gestión de una amplia variedad de tareas desde teléfonos móviles o dispositivos de tipo PDA (*Personal Digital Assistant*) y con provisión de calidad de servicio para aplicaciones de voz, y (c) WPAN 802.15.4 (*Low Rate WPAN*) [IEEE 802 03b, IEEE 802 06] concebidas para aplicaciones médicas e industriales de muy bajo consumo, sin requisitos estrictos de tasa de transferencia y de QoS.

En la actualidad, la tecnología más utilizada en las WPAN es Bluetooth [Shinde05, Záruba04]. Tal y como se apunta en [Záruba04], la característica clave que ha propiciado la viabilidad de las WPAN basadas en tecnologías Bluetooth ha sido el bajo coste de los módulos Bluetooth, integrado ya en una alta variedad de dispositivos comerciales.

Así, desde la aparición en el mercado de los primeros dispositivos Bluetooth en el año 2000 (teléfonos móviles, auriculares o *headsets* y *PC Cards*) ha ido aumentando la variedad de dispositivos dotados con tecnología Bluetooth, llegándose a incorporar 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 [Wang07].

En este sentido, existe un factor determinante que contribuye a la expansión de cualquier sistema, y que ha impulsado el éxito de Bluetooth frente a las tecnologías competidoras, incluso en áreas de aplicación para las que inicialmente no se concibió esta tecnología, como es el caso de la telemedicina. Este factor es la

interoperatividad. Con el fin de garantizar dicha interoperatividad entre los dispositivos Bluetooth de distintos fabricantes, la especificación Bluetooth establece distintos perfiles [Bluetooth SIG01f], en los que se definen 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. Por ejemplo, la mayoría de los dispositivos Bluetooth de interacción (teclados, mandos de consolas, e incluso teléfonos móviles) se caracterizan por implementar el perfil HID (*Human Interface Device*) mientras que, prácticamente todos los demás, incluyendo los sensores médicos comerciales existentes en la actualidad, de fabricantes distintos (Nonin Medical [Nonin Medical], CorScience [CorScience] y A&D Medical [A&D Medical]), implementan el perfil SPP (*Serial Port Profile*).

Sin embargo, lo realmente sorprendente no es tanto la alta difusión comercial lograda, sino el hecho de que la especificación del sistema Bluetooth, al tiempo que es objeto de investigación, continúa actualizándose y ampliando, diez años después de sus orígenes y teniendo en cuenta el estado de la técnica.

Una de las áreas más extensas de aplicación de las WPAN basadas en Bluetooth, en la que se centra la investigación actual, es la de telemedicina. La razón es que el uso de estándares de interoperatividad contribuye a aumentar el grado de aceptación y de difusión de sistemas de esta naturaleza, ya que tal y como apunta Warren en [Warren04], los estándares, aplicados adecuadamente al diseño de sistemas médicos, tienen el potencial de reducir el coste de los sistemas de monitorización al tiempo que mejoran la atención a las necesidades de los pacientes. De hecho, Warren en la serie de trabajos presentados en [Warren03, Warren04, Warren05, Yao05] elige Bluetooth como un estándar de

telemetría, considerando incluso que es una tecnología idónea para sistemas reconfigurables y «vestibles» (*wearables*), por el hecho de que soporta *scatternets*, por el consumo reducido, por su coste, por el uso de técnicas de transmisión de espectro ensanchado, servicios de seguridad, así como por el soporte comercial existente.

La importancia de que una profusa producción científica abogue por el uso de Bluetooth en el ámbito de la telemedicina radica en el hecho de que está confluyendo con los esfuerzos de estandarización llevados a cabo para la definición de un nuevo perfil Bluetooth específico de dispositivos médicos: MDP (*Medical Device Profile*). El perfil MDP, actualmente en estado de desarrollo, aumentará el grado de interoperatividad, ya que aunque SPP garantiza la interoperatividad a nivel de comunicación, la información médica transmitida por los dispositivos tiene un formato distinto, definido por cada fabricante. Con este fin, se prevé que el perfil MDP integre el estándar ISO/IEEE 11073 [IEEE 1073 96, IEEE 1073 07] (formalmente conocido como IEEE 1073-*Medical Information Bus*) de interoperatividad entre los componentes de un sistema de monitorización médico.

Además de la interoperatividad hay que tener en cuenta que el soporte de calidad de servicio es otro requisito imprescindible para garantizar la utilización del ancho de banda de las aplicaciones móviles basadas en este tipo de WPAN. Sin embargo, hasta el momento, un número significativo de trabajos no contemplan el uso de perfiles, considerando que los datos se entregan directamente a niveles inferiores de la arquitectura Bluetooth, obviando así el *overhead* introducido por las capas superiores. Al respecto, es conveniente aclarar que esta opción no es factible en sistemas en los que se integran dispositivos

reales con un determinado perfil, normalmente SPP, como es el caso de los prototipos de sistemas de telemedicina presentados en [Moron05a, Moron05b, Moron05c, Moron07a, Moron07c].

Es precisamente el hecho de que los sensores médicos utilizados en los prototipos de sistemas de monitorización descritos en estos trabajos implementen el perfil SPP lo que ha motivado, en esta tesis, el estudio de rendimiento, en términos de retardo extremo a extremo, para este perfil.

También se ha querido comparar el comportamiento del perfil SPP con el del perfil PAN (*Personal Area Network*), al que se le augura una futura expansión debido a que permite el encapsulamiento de tráfico IP sobre Bluetooth, aunque actualmente no esté soportado por ningún dispositivo comercial.

Concretamente, el objetivo de esta tesis es estudiar el rendimiento de una WPAN Bluetooth, en términos del retardo extremo a extremo y del *throughput*, sin prescindir del uso de perfiles y evaluando el efecto de la configuración de la calidad de servicio.

Con esta finalidad, en una primera fase, se comienza proponiendo un modelo de retardo para los perfiles PAN y SPP, y se lleva a cabo una comparación del rendimiento, considerando el *overhead* agregado por los protocolos que intervienen en la transmisión de la información de usuario, y el «desajuste por segmentación» —*segmentation mismatch*—, que puede producirse cuando no hay coincidencia en el máximo tamaño de los datos manejados entre capas adyacentes de la arquitectura.

En una segunda fase, y teniendo en cuenta que SPP es el perfil soportado por la amplia mayoría de los dispositivos médicos comerciales existentes hoy

día, se extiende el modelo propuesto para SPP, con el fin de contemplar el efecto de la gestión de calidad y el incremento en el retardo de transmisión debido a las pérdidas.

Finalmente, en una tercera fase, se han llevado a cabo medidas con dispositivos reales para contrastar y corregir los modelos propuestos.

La presente memoria, en la que se expone el estudio realizado, consta de seis capítulos, en cada uno de los cuáles se abordan los siguientes contenidos:

En el capítulo 2 se presenta una breve introducción a la tecnología Bluetooth.

En el capítulo de revisión bibliográfica, capítulo 3, se expone un compendio de aquellos trabajos relacionados con el soporte de calidad en redes Bluetooth, pero que generalmente no consideran la utilización de perfiles.

En el capítulo 4 se presenta el estudio teórico realizado, estructurado como se indica a continuación:

- Estudio y comparación del rendimiento de los perfiles SPP y PAN (sección 4.1).
- Estudio del efecto de la configuración de calidad en el caso del perfil SPP (sección 4.2).
- Estudio, para SPP, del efecto de las pérdidas debido a la interferencia (sección 4.3).

En el capítulo 5 se describen las medidas de retardo realizadas con dispositivos Bluetooth 1.1, comparando los resultados con los obtenidos teóricamente de acuerdo con el siguiente esquema:

-
- Comparación del retardo teórico y medido: Perfiles SPP y PAN (sección 5.2).
 - Efecto de la configuración de la calidad de servicio (sección 5.3).
 - Efecto de las retransmisiones (sección 5.4).

El capítulo 5 se complementa con el apéndice A, en el que se presenta la repetición de un subconjunto de medidas efectuadas con dispositivos Bluetooth 1.2 y Bluetooth 2.0, que permiten validar el modelo para dichas versiones.

Finalmente, en el capítulo 6, se resumen las conclusiones extraídas y se plantean las líneas futuras de investigación.

Capítulo 2

Fundamentos Teóricos de Bluetooth

El grupo Bluetooth SIG (*Special Interest Group*), fundado en el año 1998 por cinco compañías, desarrolló la especificación de la tecnología Bluetooth, cuya versión 1.0, que se publicó en el año 1999, fue actualizada a la versión 1.1 en el año 2001. Esta versión, en el año 2002, fue parcialmente adoptada por el IEEE (*Institute of Electronic and Electrical Engineering*) bajo el estándar IEEE 802.15.1-2002 [IEEE 802 02] (véase la sección 2.1).

La especificación del sistema Bluetooth 1.1, en la que se basa el estándar IEEE 802.15.1-2002, consta de los siguientes documentos:

1. *Bluetooth Core Specification Volume 1* [Bluetooth SIG01c]: Describe y especifica la arquitectura de protocolos Bluetooth.
2. *Bluetooth Profiles Specification Volume 2* [Bluetooth SIG01f]: Especifica los perfiles básicos y su relación con la arquitectura de protocolos.

3. *Bluetooth Assigned Numbers Bluetooth* [Bluetooth SIG01a]: Define códigos e identificadores del estándar Bluetooth.
4. *Bluetooth continuous variable slope delta (CVSD) encoded test signal Bluetooth* [Bluetooth SIG01b]: Proporciona la señal de prueba utilizada para validar el codificador de voz del estándar Bluetooth.
5. *Bluetooth Personal Area Networking Profile* [Bluetooth SIG01e]: Especifica el perfil para el establecimiento de redes PAN.
6. *Bluetooth Network Encapsulation Protocol (BNEP)* [Bluetooth SIG01d]: Define el protocolo BNEP utilizado en el perfil PAN.

Posteriormente, tras la publicación de la versión 1.2 de la especificación «*Bluetooth Core Specification*» [Bluetooth SIG03a] en el año 2003, también se efectuó una revisión del estándar 802.15.1, dando lugar en el año 2005 a la versión 802.15.1-2005 [IEEE 802 05].

En el año 2004, el grupo Bluetooth SIG publicó la versión Bluetooth 2.0 + EDR (*Enhanced Data Rate*) [Bluetooth SIG04] y, recientemente, en Noviembre de 2007, se ha hecho pública la siguiente versión Bluetooth 2.1 + EDR [Bluetooth SIG07].

El estudio que se expone en el presente trabajo se ha llevado a cabo con dispositivos Bluetooth 1.1. Por este motivo, en la sección 2.1, con la intención de facilitar el seguimiento de los modelos teóricos propuestos, se introducen los conceptos fundamentales de la tecnología en el ámbito de la especificación 1.1 [Bluetooth SIG01f]. A continuación, en las secciones 2.2 y 2.3, se presentan las modificaciones más destacables introducidas, respectivamente, en Bluetooth 1.2 y en Bluetooth 2.0 + EDR. Al respecto, es conveniente aclarar que tanto

la especificación Bluetooth 1.2 como Bluetooth 2.0 + EDR son totalmente compatibles con la especificación Bluetooth 1.1, y que los cambios incorporados en cada actualización corresponden a extensiones de la funcionalidad definida en Bluetooth 1.1.

Finalmente, en la sección 2.4 se realiza una breve introducción de los perfiles definidos en [Bluetooth SIG01f], dedicando más atención a los perfiles objeto de estudio: SPP y PAN.

2.1. Bluetooth 1.1

La finalidad con la que se concibe la tecnología Bluetooth es reemplazar el cable de conexión entre dispositivos electrónicos, fijos o móviles, por un enlace radio de corto alcance (10–100 m), robusto frente a desvanecimientos e interferencias y que permita la transmisión fiable de voz y datos. Para ello se utiliza una técnica de espectro ensanchado, denominada *frequency hopping*, que consiste en ir seleccionando de forma pseudo–aleatoria la frecuencia a la que hay que sintonizar el transceptor radio.

El sistema Bluetooth proporciona conexiones punto a punto (entre dos unidades Bluetooth) o punto-multipunto. Cuando dos o más unidades comparten el mismo canal se constituye una *piconet*, en la que un dispositivo actúa como maestro y los restantes como esclavos.

En una *piconet* puede haber hasta siete esclavos activos. No obstante, el llamado modo *PARK* permite que haya muchos más esclavos participando en la *piconet* que, aunque no están activos, sí están sincronizados con la misma.

Varias *piconets* con áreas de cobertura solapadas dan lugar a lo que se denomina una *scatternet*. Se forma una *scatternet* cuando un esclavo se conecta a más de una *piconet* o bien cuando un maestro participa en otra *piconet* como esclavo. Las *piconets* que coexisten en una *scatternet* no están sincronizadas en frecuencia. Esto es, cada *piconet* tiene su propia secuencia de salto.

La arquitectura de protocolos consta de los siguientes componentes:

- El subsistema *Bluetooth Controller* o Controlador Bluetooth, conocido como módulo Bluetooth, del que forman parte los tres niveles inferiores: *Radio Frequency* (RF), *Baseband* (BB) y *Link Manager* (LM).
- El subsistema *Host*, al que pertenecen el nivel L2CAP (*Logical Link Control and Adaptation Protocol*) y las restantes capas superiores.

No obstante, la arquitectura que se presenta en la figura 2.1 responde a una implementación muy habitual, en la que se incluye una interfaz opcional de comunicación física denominada HCI (*Host Controller Interface*), que está integrada por tres componentes: *HCI Driver*, *HCTL* (*Host Controller Transport Layer*) y *HCI Firmware*. Aunque esta interfaz es opcional, su utilización permite garantizar la interoperatividad entre subsistemas Bluetooth (*Host* y *Bluetooth Controller*).

En relación con el estándar 802.15.1, cabe destacar que en su definición se incorporan, como nivel MAC (*Medium Access Control*), las capas L2CAP, LM y BB, y como nivel físico (PHY), la capa RF, quedando fuera del ámbito del estándar 802.15.1 los protocolos situados por encima del nivel L2CAP. La figura 2.2 ilustra esta correspondencia definida en la primera sección del estándar

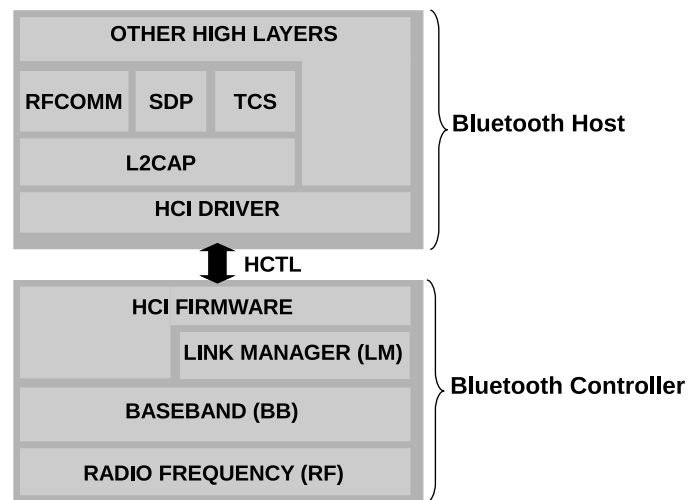


Figura 2.1: Arquitectura de protocolos Bluetooth 1.1

802.15.1-2002 [IEEE 802 02], así como la equivalencia con el modelo OSI (*Open Systems Interconnection*).

Las funciones de los niveles BB, RF, LM, HCI y L2CAP se describen someramente en los sucesivos apartados de esta sección.

2.1.1. Nivel RF (*Radio Frequency*)

Las características que definen al subsistema radio son las siguientes:

- Opera en la banda ISM (*Industrial, Scientific and Medical*) a 2,4 GHz, concretamente en la banda 2,4000–2,4835 GHz para la que se definen 79 canales espaciados 1 MHz a las frecuencias $f = 2402 + k$ MHz con $k = 0, \dots, 78$ y en la banda 2,4465–2,4835 GHz en la que se sitúan 23 canales RF en las frecuencias $f = 2454 + k$ MHz con $k = 0, \dots, 22$.

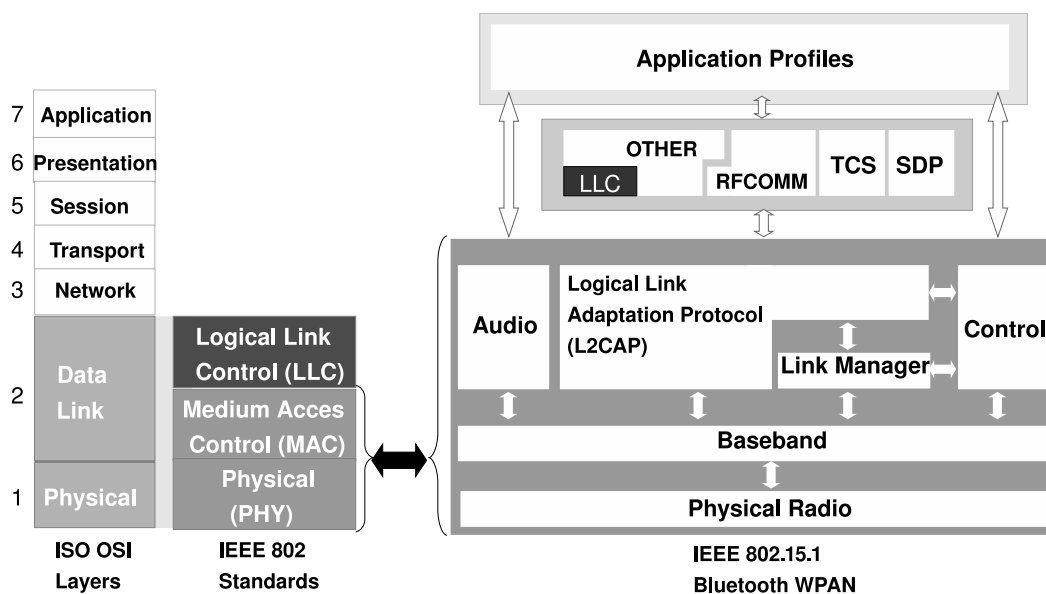


Figura 2.2: Correspondencia entre la arquitecturas Bluetooth, IEEE 802 y modelo OSI (fuente: estándar 802.15.1-2002 [IEEE 802 02])

Mediante la utilización de una banda u otra, la especificación Bluetooth 1.1 contempló el hecho de que la banda de operación elegida, ISM a 2,4 GHz, de libre uso a nivel mundial, inicialmente no dispusiera en todos los países del mismo rango de frecuencias.

- Transmisión a 1 Mbit/s.
- Empleo de modulación GFSK (*Gaussian Frequency Shift Keying*), para minimizar la complejidad del transceptor.
- Cambio de la frecuencia de operación a una velocidad de 1600 saltos/s (*frequency hopping*).
- Consumo de potencia que, dependiendo de la clase del dispositivo Bluetooth, está limitado a los valores que se indican en la tabla 2.1.

Clase	Potencia	Alcance
1	100 mW (20 dBm)	100 m
2	2,5 mW (4 dBm)	10 m
3	1 mW (0 dBm)	10 cm - 1 m

Tabla 2.1: Clases de dispositivos Bluetooth

2.1.2. Nivel BB (*Baseband*)

El nivel BB realiza el control de la temporización de transmisión y recepción, determina la secuencia de salto que define el canal de la *piconet*, genera todos los tipos de claves existentes y realiza el procesamiento de los datos de las capas superiores (L2CAP y LM) componiendo (y descomponiendo) los paquetes en los que se transmiten (y reciben) estos datos. El nivel BB, además, proporciona, para determinados tipos de paquetes, mecanismos de control de errores y de flujo.

En relación con este nivel, los conceptos de canal físico y de enlace físico se definen respectivamente en los apartados 2.1.2.1 y 2.1.2.2. En el apartado 2.1.2.3 se resume la máquina de estados y los modos de operación definidos en la especificación Bluetooth. Por último, en el apartado 2.1.2.4 se presenta la estructura del paquete Bluetooth y los tipos de paquetes utilizados para la transmisión de datos de los niveles superiores sobre los enlaces físicos proporcionados por el nivel BB.

2.1.2.1. Canal Físico

El canal está representado por una secuencia de salto pseudo-aleatoria de 79 ó 23 canales RF, es decir, saltos consecutivos se corresponden a frecuencias RF

diferentes. Esta secuencia es única para cada *piconet* ya que se determina a partir de la dirección Bluetooth y del reloj del maestro.

El canal está dividido en «ranuras» temporales o *time slots* de $625 \mu s$ de duración. Los *slots* se numeran de acuerdo con el reloj del maestro de la *piconet*, comenzando en 0 y finalizando en $2^{27} - 1$. Por este motivo, se dice que los dispositivos integrantes de una *piconet* están sincronizados al canal en tiempo y frecuencia.

Maestro y esclavos alternan sus transmisiones siguiendo un esquema TDD (*Time Division Duplex*), de forma que el maestro debe transmitir en los *slots* pares y los esclavos en los impares, siempre y cuando hayan sido direccionados por el maestro en el *slot* anterior.

A cada *slot* corresponde una frecuencia RF de la secuencia de salto. Sin embargo, en el caso de que el paquete transmitido ocupe más de un *slot*, la frecuencia no cambiará mientras dure el paquete y la frecuencia de transmisión del próximo paquete será la correspondiente al *slot* en el que se envía. Por ejemplo, en la figura 2.3 se observa que si un paquete de tres *slots* se transmite a la frecuencia $f(k)$, la siguiente frecuencia a emplear será $f(k + 4)$ y no $f(k + 2)$. Asimismo, si el paquete ocupa cinco *slots*, la figura muestra que el paquete que se envía a continuación comienza en la frecuencia $f(k + 6)$.

2.1.2.2. Enlaces físicos

Entre dos dispositivos Bluetooth pueden establecerse dos tipos de enlaces físicos:

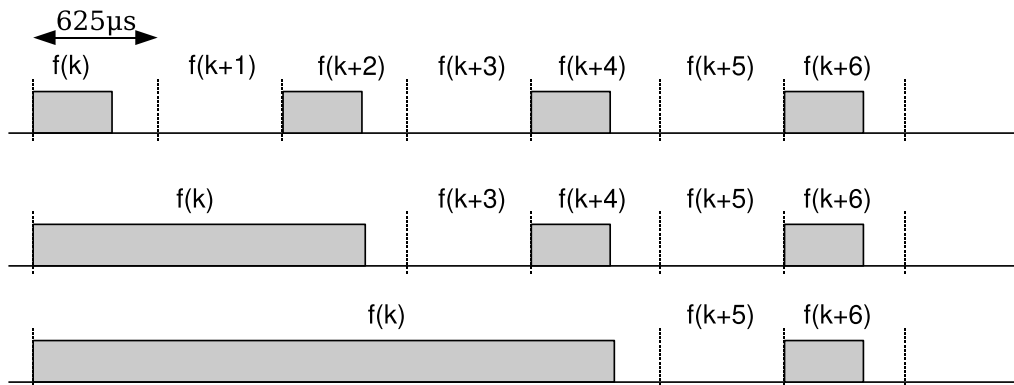


Figura 2.3: Esquema de transmisión de paquetes *multi-slot* (fuente: parte B de la especificación [Bluetooth SIG01c])

- *SCO (Synchronous Connection-Oriented)*: Es un enlace punto a punto, simétrico y orientado a conexión cuyo objetivo es soportar la transmisión de información síncrona, normalmente voz a 64 kb/s.

El enlace SCO reserva *slots* y, por tanto, puede considerarse como una conexión de conmutación de circuitos entre el maestro y el esclavo.

- *ACL (Asynchronous Connection-Less)*: Es un enlace punto a multipunto entre el maestro y todos los esclavos de la *piconet*. Este tipo de enlace, que se considera como una conexión de conmutación de paquetes, permite el envío de información asíncrona e isócrona, aprovechando la capacidad que queda libre en los *slots* no reservados para enlaces SCO. Puede soportarse una tasa de tráfico asimétrica máxima de 723,2 kb/s (y hasta 57,6 kb/s en el sentido de comunicación opuesto), o de 433,9 kb/s por cada uno de los sentidos de comunicación.

Una unidad que esté funcionando como maestro sólo puede tener establecido un enlace ACL con cada uno de los esclavos y hasta tres enlaces SCO,

independientemente de que éstos sean con un mismo esclavo o con distintos. Sin embargo, en el caso de un esclavo, el máximo número de enlaces SCO se ve reducido a dos cuando se establecen con maestros diferentes, ya que es necesario un tiempo de guarda para conmutar de una *piconet* a otra.

2.1.2.3. Estados y modos de operación

El diagrama presentado en la figura 2.4 muestra los estados definidos para el nivel BB. En éste se observa que hay dos estados principales, *STANDBY* y *CONNECTION*, y que la transición de *STANDBY* a *CONNECTION* se realiza pasando por los subestados intermedios: *inquiry*, *inquiry scan*, *inquiry response*, *page*, *page scan*, *slave response* y *master response*.

El estado de reposo de una unidad Bluetooth es *STANDBY*. Se trata de un estado de bajo consumo en el que solamente funciona el reloj del sistema, y del que una unidad Bluetooth sale para llevar a cabo un procedimiento de *INQUIRY* o de *PAGE*.

El procedimiento de *INQUIRY* lo desencadena una unidad Bluetooth cuando quiere detectar la presencia de otras unidades en su zona de cobertura, o averiguar la dirección de un determinado dispositivo o grupo de dispositivos. El proceso consiste en difundir mensajes que contienen, en el primer caso, un código genérico, susceptible de ser reconocido por cualquier unidad Bluetooth, y en el segundo caso, el código definido para la clase de dispositivo que se desea que responda.

Al iniciarse el citado procedimiento, el nivel BB pasa al subestado *inquiry* y permanecerá en él transmitiendo continuamente los mensajes mencionados,

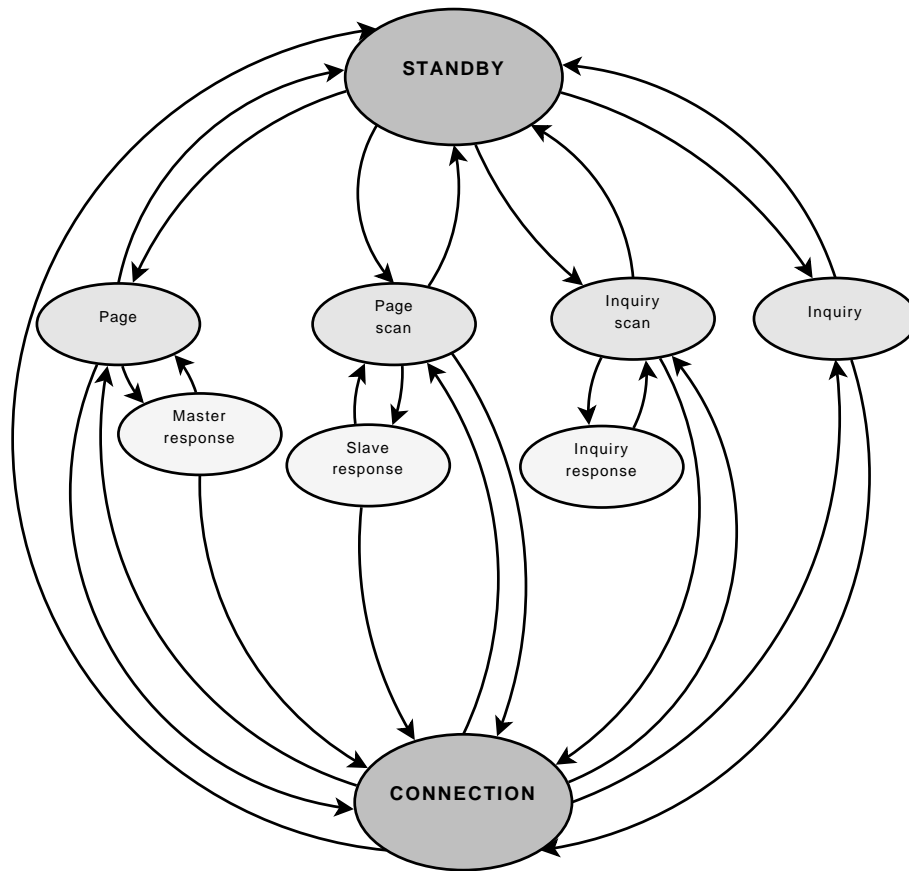


Figura 2.4: Diagrama de estados definido en Bluetooth 1.1 (fuente: parte B de la especificación [Bluetooth SIG01c])

hasta que se agote el tiempo de espera o hasta que el nivel LM lo solicite, porque considere que se han recopilado suficientes respuestas.

Para que una segunda unidad Bluetooth (unidad B) pueda detectar los mensajes de *inquiry* enviados por la primera (unidad A), debe encontrarse en el subestado *inquiry scan*. Cuando la unidad B recibe un mensaje de *inquiry*, debe pasar al subestado *inquiry response* para responder.

Una vez que el dispositivo A, como resultado de un procedimiento de *INQUIRY*, conoce la dirección Bluetooth, *BD_ADDR* (*Bluetooth Address*), y la información de sincronía (reloj), entre otros parámetros, de las unidades que se

encuentran en su entorno, puede querer establecer una conexión con alguna de ellas. Para ello, se ha definido en la norma el procedimiento de *PAGE*. La unidad que inicia este procedimiento, A, desempeña el papel de maestro mientras que el dispositivo con el que se va a intentar establecer la conexión, B, actuará como esclavo. Para el primero, el subestado de partida es *page* y para el segundo, *page scan*.

En el subestado *page*, el maestro transmite el mensaje de *PAGE* en la primera y en la segunda mitad de cada *slot* de transmisión, hasta que el esclavo contesta o hasta que se excede la temporización correspondiente. Cuando el esclavo responde a dicho mensaje, pasa al subestado *slave response* y cuando el maestro recibe la respuesta, entra en el subestado *master response*. En este subestado, el maestro tiene que enviar al esclavo un paquete de control con todos los parámetros necesarios para que éste pueda sincronizarse con la *piconet*. El esclavo aguardará la llegada de este paquete mientras no expire el temporizador correspondiente a este subestado. Si lo recibe, deberá confirmarlo antes de transitar al estado *CONNECTION* y, si no, volverá al subestado *page scan*. Por su parte, el maestro tiene que retransmitir el mencionado paquete hasta que reciba respuesta del esclavo o hasta que se exceda el tiempo máximo de espera. En este último caso, se volverá al subestado *page*. Si, por contra, antes de que se cumpla este plazo, la confirmación llega a la unidad A, está enviará a B un paquete por el canal de la *piconet*, de forma que, para dar por concluido el establecimiento de la conexión, es condición indispensable que el esclavo responda antes de un número determinado de *slots*. De no ser así, maestro y esclavo volverán al punto de partida: subestados *page* y *page scan*.

En el estado *CONNECTION* se distinguen los siguientes modos de funcio-

namiento:

- *ACTIVE*: Con el objeto de identificar a los esclavos participantes en la *piconet*, el maestro asigna a cada esclavo una dirección que será válida mientras se encuentre en este modo de operación. Esta dirección, denominada AM_ADDR (*Active Member Address*) se incluye en la cabecera de los paquetes intercambiados en los dos sentidos de comunicación (véase la figura 2.5 (a) del apartado 2.1.2.4). Se reserva el valor nulo para identificar los mensajes de difusión (*broadcast*) dirigidos a todos los esclavos de la *piconet*.

El maestro planifica las transmisiones a los esclavos teniendo en cuenta que, si no hay peticiones de datos de capas superiores, debe enviar regularmente paquetes de control para permitir que los esclavos sigan sincronizados.

Los esclavos, en cada *slot* de transmisión del maestro (*slots* pares), comprobarán si han sido direccionados, de forma que, en caso negativo, si el paquete recibido ocupa más de un *slot*, podrán permanecer con el transceptor apagado hasta que concluya la recepción de dicho paquete.

- *SNIFF*: El objetivo de este modo es permitir que una unidad que está funcionando como esclavo pueda reducir la actividad de escucha. Para ello, el maestro debe comenzar la transmisión de los paquetes destinados a dicho dispositivo en *slots* regularmente espaciados, de acuerdo con la temporización negociada con el esclavo.
- *HOLD*¹: Con el fin de reducir el consumo o de liberar capacidad para poder

¹El soporte de los enlaces SCO no se ve afectado.

llevar a cabo otras actividades, tales como un procedimiento de búsqueda de otros dispositivos, el establecimiento de una conexión o la participación en otra *piconet*, se interrumpe el envío de paquetes ACL durante un tiempo, el cual ha sido negociado a nivel de LM.

- *PARK*: Es el modo en el que entra una conexión con un esclavo que no necesita participar activamente en la *piconet*, pero que quiere continuar sincronizado. El esclavo perderá la dirección de miembro activo (*AM_ADDR*), que el maestro le asignó al establecer la conexión, pero en su lugar recibe otros identificadores:
 - *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.

Para finalizar, en relación con la figura 2.4, hay que añadir que el funcionamiento en el estado *CONNECTION* se puede simultanear con la realización de procedimientos de *INQUIRY* (o de *INQUIRY SCAN*) y de *PAGE* (o de *PAGE SCAN*).

2.1.2.4. Estructura y tipos de los paquetes BB

Los paquetes especificados en el nivel BB se clasifican en cuatro grupos. El grupo 1 engloba los cuatro paquetes de control comunes a ambos tipos de

Grupo	N.º de slots	Enlace SCO	Enlace ACL
1	1	NULL	NULL
	1	POLL	POLL
	1	FHS	FHS
	1	DM1	DM1
2	1	No definido	DH1
	1	HV1	No definido
	1	HV2	No definido
	1	HV3	No definido
	1	DV	No definido
	1	No definido	AUX1
3	3	No definido	DM3
	3	No definido	DH3
4	5	No definido	DM5
	5	No definido	DH5

Tabla 2.2: Tipos de paquetes Bluetooth

enlaces (ACL y SCO): NULL, POLL, FHS y DM1, y los grupos 2, 3 y 4, los paquetes que ocupan 1, 3 y 5 slots, respectivamente. La tabla 2.2 muestra la clasificación descrita, especificando para cada enlace los paquetes definidos en la versión 1.1.

La norma define cuatro paquetes para el enlace SCO: HV1, HV2, HV3 y DV. Los paquetes HV1, HV2, HV3 transportan exclusivamente 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. El paquete DV permite el envío simultáneo de 10 octetos de voz y de hasta 9 octetos datos, éstos últimos, codificados con FEC $2/3$.

Para el enlace ACL se especifican un total de siete tipos de paquetes ACL: DM1, DH1, DM3, DH3, DM5, DH5 y AUX1. Los paquetes ACL para los que se

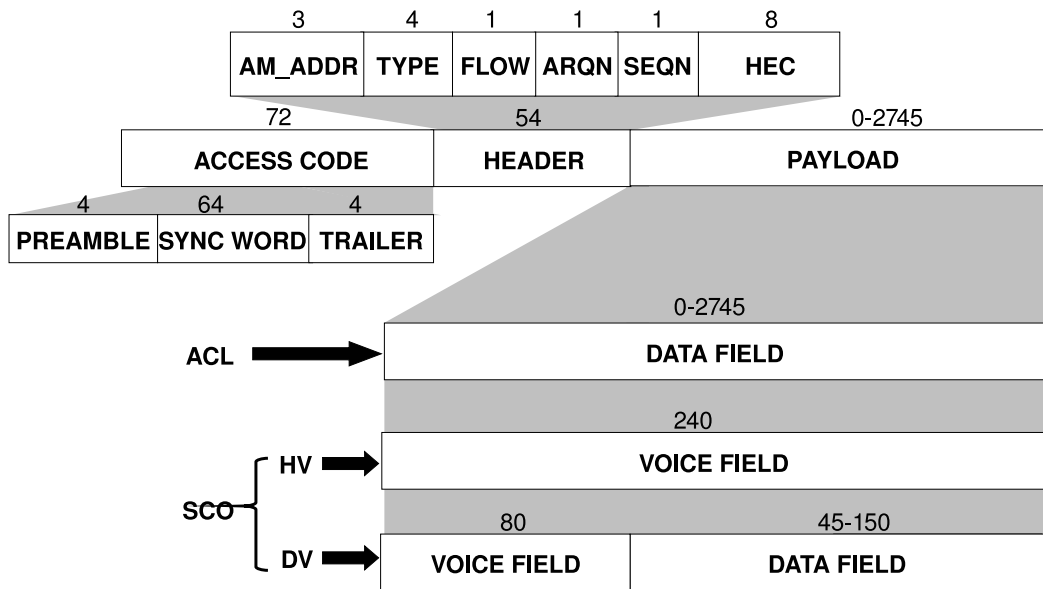
proporciona control de errores (véase el apartado 2.1.2.5) son los de tipo DH (*Data High-Rate*) y DM (*Data Medium-Rate*), es decir, todos menos AUX1. En los paquetes DM se utiliza, además, codificación FEC $2/3$ como mecanismo de protección de errores.

Un paquete consta de tres campos, integrados por el número de bits que se indica en la figura 2.5 (a):

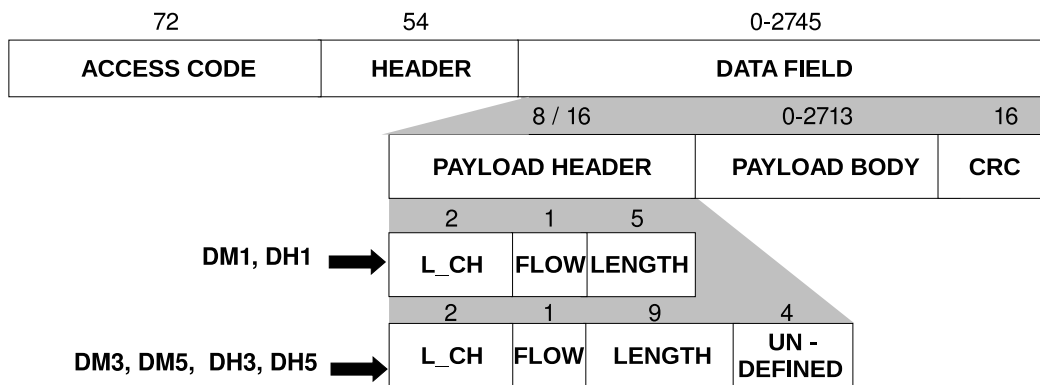
- Código de Acceso (*Access Code*): Existen tres tipos de códigos:
 - DAC (*Device Access Code*): Se utiliza en el procedimiento de establecimiento de conexión.
 - IAC (*Inquiry Access Code*): Se emplea en el procedimiento de búsqueda de dispositivos.
 - CAC (*Channel Access Code*): Permite identificar la *piconet* en la que se está intercambiando un paquete ACL o SCO.

El CAC, con una longitud total de 72 bits, consta de tres campos: preámbulo (*preamble*), palabra de sincronización (*sync word*) y cola (*trailer*). Sin embargo, en los códigos DAC e IAC se omite el último campo (cola).

- Cabecera (*Header*): Contiene información de control, estructurada en los siguientes campos, y protegida con codificación FEC $1/3$.
 - AM_ADDR (3 bits): Dirección del esclavo al que va dirigido el paquete.
 - TYPE (4 bits): Código que identifica el tipo de paquete que se está transmitiendo.



(a) Formato general de los paquetes Bluetooth



(b) Formato del campo de datos (*Data Field*)

Figura 2.5: Estructura de los paquetes Bluetooth y formato del campo de datos (el tamaño de los campos está indicado en bits)

- 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 notificar a su homólogo la integridad del paquete recibido o, en su defecto, para solicitar la retransmisión (véase el apartado 2.1.2.5).
 - SEQN (1 bit): Mediante este bit, el nivel BB indica si el paquete que se transmite 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*), que transporta la información de las capas superiores. En el caso de los paquetes ACL recibe la denominación de campo de datos (*Data Field*) y su estructura, mostrada en la figura 2.5 (b), consta de:
- Una cabecera (*payload header*), que tendrá una extensión de 8 o de 16 bits, dependiendo de que el paquete ocupe uno o más *slots*. 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 campo de datos propiamente dicho (*payload body*), que transporta, según indique el campo L_CH de la cabecera, información del nivel LM (L_CH=11) o un fragmento, inicial (L_CH=10) o intermedio (L_CH=01), de un mensaje L2CAP.
 - El código CRC (*Cyclic Redundancy Check*): Código de verificación de errores de los datos, que no se incluye en el paquete AUX1.

En los paquetes HV el *payload* contiene únicamente el campo de voz (*Voice Field*), ya que no existe la cabecera (*payload header*), ni el código CRC. Sin embargo, en el paquete DV, el *payload*, tal y como se observa en la figura 2.5 (a), consta de un campo de voz y de un campo de datos.

En la tabla 2.3, para cada tipo de paquete ACL, se indica el tamaño del campo de datos (*payload body*), así como si la información se protege, o no, mediante codificación FEC 2/3, y si se incluye el código de comprobación CRC. Obsérvese que el paquete AUX1 puede transportar 2 octetos más que el paquete DH1, por el hecho de que no incluye los dos octetos correspondientes al CRC. Nótese que, a partir de la máxima longitud de los paquetes, puede deducirse que éstos no se extienden hasta el final del último (o único) *slot*. Por ejemplo, un paquete DH1 consta de 366 bits, ocupando tan solo un 58,66 % del *slot*.

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

Tabla 2.3: Tipos de paquetes ACL

Finalmente, es conveniente aclarar que, además de los paquetes de control comunes, citados en el grupo 1 de la tabla 2.2, existe un quinto paquete de control, el paquete ID. A continuación, se describen los cinco paquetes de control definidos en la especificación:

- *Paquete ID*: Consta únicamente del código de acceso (DAC o IAC), y es empleado en los procedimientos de INQUIRY y de PAGE.
- *Paquete FHS*: Es utilizado en los procedimientos de INQUIRY y de PAGE. Tiene un formato específico, que no coincide con el descrito en la figura 2.5 (para más información véase el apartado 4.4.1.4 de la parte B de la especificación [Bluetooth SIG01c]).
- *Paquete NULL*: Consta de los dos primeros campos: código de acceso y cabecera. Se utiliza para transmitir información de control de errores y de flujo relativa a tráfico ACL, y no tiene que ser confirmado.
- *Paquete POLL*: Al igual que el paquete NULL, sólo contiene los campos código de acceso y cabecera. Es empleado por el maestro de la *piconet* cuando habiendo transcurrido el intervalo de *poll*, o *poll interval*, (véase el apartado 2.1.3), le corresponde direccionar a un esclavo para el que no tiene datos que transmitir. El esclavo que recibe un paquete POLL está obligado a confirmar al maestro su recepción.
- *Paquete DM1*: Este paquete se incluye en el primer grupo debido a que, además de transportar datos de usuario, se utiliza para el envío de información de control de cualquier tipo de enlace, pudiendo, por tanto, incluso interrumpir la transmisión de información síncrona.

2.1.2.5. Corrección de Errores

En el nivel BB se definen tres mecanismos de corrección de errores:

- FEC 1/3: Consiste en repetir tres veces cada uno de los bits del campo a codificar. Se utiliza para la protección de los bits de la cabecera y del campo de voz del paquete HV1.
- FEC 2/3: Código de Hamming (15,10) recortado (*shortened*)², que añade 5 bits de redundancia por cada 10 bits de información. Se utiliza para proteger la información del campo de datos (*Data Field*) en los paquetes DM, permitiendo la corrección de un error y la detección de dos errores por cada bloque de 15 bits.
- Esquema ARQ (*Automatic Repeat reQuest*): Bluetooth utiliza un esquema ARQ no numerado como mecanismo adicional del control de errores en los paquetes DH, DM y DV. De acuerdo con este esquema, el nivel BB de una unidad que recibe un paquete DH, DM o DV confirma al extremo origen, positiva (*ACK, Acknowledge*) o negativamente (*NAK, Negative Acknowledge*), la correcta recepción de ese último paquete. El esclavo responderá en el *slot* esclavo-maestro siguiente al *slot* en el que finaliza la transmisión del maestro. Sin embargo, el maestro podrá posponer el envío de la confirmación a la próxima vez que direcciona al esclavo del que ha recibido el paquete. Se enviará confirmación positiva (ARQN=1) en caso de que no falle la verificación del HEC ni del CRC. De lo contrario, se indicará el estado de error al extremo remoto desactivando el bit ARQN (ARQN=0) en la cabecera del siguiente paquete que se transmita. Como respuesta a la confirmación negativa, el paquete se estará retransmitiendo hasta que se reciba una confirmación positiva o hasta que se exceda el temporizador *flushTO* (véase el apartado 2.1.3).

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

2.1.3. Nivel LM (*Link Manager*)

Este nivel es responsable del establecimiento y control de los enlaces físicos ACL y SCO entre dispositivos Bluetooth. La comunicación entre niveles homólogos se realiza siguiendo el protocolo LMP (*Link Manager Protocol*) que define las transacciones necesarias para llevar a cabo las funciones de gestión de los enlaces. La transmisión de mensajes LMP tiene prioridad sobre el tráfico de la capa superior L2CAP, aunque puede retardarse debido a la retransmisión de paquetes BB individuales correspondientes a un fragmento de un mensaje L2CAP.

Entre otras funciones, el nivel LM es el responsable de proporcionar, al nivel L2CAP, capacidad de configuración de calidad de servicio para los enlaces ACL, permitiéndole reservar ancho de banda y controlar la latencia de la transmisión. En concreto, el nivel LM negocia con la entidad homóloga 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. El valor mínimo del intervalo de *poll* es de 2 *slots* (1,25 ms) y el valor especificado por defecto es de 40 *slots* (25 ms).

En la especificación se indica que una unidad que esté actuando como maestro sólo está obligada a comunicarse con cada esclavo de la *piconet* una vez cada T_{poll} *slots*.

El intervalo de *poll* se garantiza en el modo activo (*ACTIVE*), excepto cuando se producen colisiones con la realización de los procedimientos asociados a los subestados *page*, *page scan*, *inquiry* e *inquiry scan*.

2.1.4. Nivel HCI (*Host Controller Interface*)

El nivel HCI, constituido, como ya se explicó en la sección 2.1, por el *HCI Driver*, *HCTL* y *HCI Firmware*, proporciona a los niveles integrados en el *Host* una interfaz estándar de acceso a las capacidades del módulo Bluetooth. La especificación de esta interfaz de comunicación define:

- Comandos HCI: Son generados por el *Host* para consultar registros *hardware* o iniciar una tarea, como por ejemplo, la búsqueda de dispositivos, el establecimiento de una conexión, o la configuración de parámetros de una conexión.
- Eventos HCI: Son generados por el *Bluetooth Controller* en respuesta a los comandos HCI, o bien, para notificar la ocurrencia de un evento.
- El formato de los paquetes utilizados para la transferencia de datos ACL y SCO entre el *Host* y el *Bluetooth Controller*.
- El mecanismo de control de flujo usado en la comunicación entre el *Host* y el *Bluetooth Controller*.

Además, para que ambos subsistemas, *Host* y *Bluetooth Controller*, se abstraigan de la interfaz física que los conecta, también se especifica la capa HCTL para RS-232, UART y USB.

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

Las funciones que debe implementar el nivel L2CAP son:

- 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. Esta función resulta totalmente necesaria para dar servicio a protocolos que utilicen 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): L2CAP debe permitir el intercambio de información entre dos entidades L2CAP satisfaciendo los requisitos de calidad negociados.

Los parámetros negociados en la transacción de configuración de calidad de servicio entre los niveles L2CAP están basados en una «especificación de flujo» o *flow especificacion* similar a la definida en la recomendación RFC 1363 [IETF-NWG92]. A continuación se describen dichos parámetros, así como su correspondencia con el nivel LM:

- *Service Type*: Indica el tipo de servicio que se proporciona a la entidad de nivel superior. Se distinguen dos clases de servicio: *Best Effort* y Garantizado. Para el caso por defecto, correspondiente al servicio *Best Effort*, la norma especifica que el dispositivo remoto puede optar por ignorar los parámetros de QoS, intentar satisfacerlos o garantizarlos.

- *Token Rate*: Tasa de envío expresada en *bytes* por segundo. La aplicación del nivel superior puede enviar datos continuamente a esta tasa hasta que se alcance el límite especificado por el parámetro *Token Bucket Size*. En caso de que el valor negociado corresponda al máximo ancho de banda disponible, existe un matiz que hace depender el significado de este parámetro del tipo de servicio. Así, para:

- *Best Effort*: Se ofrecerá tanto ancho de banda como sea posible.
- Servicio Garantizado: Se proporcionará el ancho de banda disponible en el momento de la petición.

El nivel LM debería asegurar que los datos son extraídos del *buffer* de transmisión y enviados a esta tasa. Adicionalmente, el nivel LM debería garantizar que el intervalo de *poll* es suficiente como para soportar esta tasa.

- *Token Bucket Size*: Este parámetro indica el tamaño del *buffer* en *bytes*. Cuando se alcanza este límite, el nivel superior debe esperar o descartar datos. En caso de que se requiera el valor máximo, al igual que en el caso anterior, la interpretación de este parámetro dependerá del tipo de servicio:
 - *Best Effort*: Se empleará un *buffer* tan grande como sea posible.
 - Servicio Garantizado: Se reservará el máximo espacio de *buffer* disponible en el momento de la petición.
- *Peak Bandwidth*: Este parámetro representa el límite de la velocidad, expresada en *bytes* por segundo, a la que las aplicaciones van a transferir los datos.

- *Latency*: El nivel LM debería garantizar que el *polling interval* toma, a lo sumo, este valor. Si el *polling interval* necesario para soportar el *Token Rate* solicitado es inferior a este valor, debería utilizarse el mínimo entre este último y el parámetro *Latency*.
- *Delay Variation*: Este parámetro indica, en microsegundos, la diferencia entre el máximo y el mínimo retardo posible que un paquete puede experimentar. Se admite que el nivel LM ignore este parámetro debido a que no se ha definido la correspondencia entre sus valores y el *polling interval*.

Es conveniente aclarar que la especificación de L2CAP impone la implementación de, al menos, el tipo de servicio *Best Effort*.

Adicionalmente, existe un parámetro relativo a la calidad de servicio cuyo valor también es notificado a la entidad homóloga en la transacción de configuración de calidad. Este parámetro es el valor, expresado en milisegundos, del temporizador *flushTO* y determina el número de retransmisiones que se realizarán a nivel BB antes de descartar un mensaje de L2CAP.

2.2. Bluetooth 1.2

En la especificación Bluetooth 1.2 las mejoras introducidas afectan a las siguientes áreas:

- Descripción de la arquitectura.
- Reducción del tiempo de establecimiento de conexiones.

- Definición de la secuencia de salto de frecuencia adaptada o AFH (*Adaptive Frequency Hopping*), como mecanismo para combatir la interferencia producida por otras tecnologías que operan también en la banda ISM.
- Inclusión de enlaces eSCO (*Extended SCO*), para la transmisión de voz a 96 kb/s, 192 kb/s y 288 kb/s.
- Definición de modos de operación en L2CAP con control de flujo y detección de errores, además de la definición del modo básico, totalmente compatible con el funcionamiento de L2CAP definido en Bluetooth 1.1.
- Capacidad de sincronización mejorada.
- Especificación de flujo de L2CAP mejorada.

Además, cabe mencionar que en Bluetooth 1.2 el transceptor radio opera ya únicamente en la banda de frecuencia 2,4000–2,4835 GHz para la que, análogamente a Bluetooth 1.1, se definen 79 canales espaciados 1 MHz a las frecuencias $f = 2402 + k \text{ MHz}$ con $k = 0, \dots, 78$.

Adicionalmente, la especificación Bluetooth 1.2 ha sido reestructurada de forma significativa para mejorar su consistencia y legibilidad. Los cambios estructurales más importantes se han llevado a cabo en los niveles BB, LM, HCI y L2CAP. El texto en estas secciones ha sido reordenado, entre otras razones, para alinear las especificaciones con la nueva arquitectura y terminología.

2.3. Bluetooth 2.0 + EDR

La especificación Bluetooth 2.0 + EDR introduce modos de funcionamiento a 2 Mbps y 3 Mbps. Para ello se definen dos tipos de modulación:

- *Basic Rate* (obligatoria), correspondiente a una tasa de transferencia de 1 Mbps.
- *Enhanced Data Rate* (opcional), que utiliza una modulación PSK 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.

2.4. Perfiles Bluetooth

Con el fin de garantizar la interoperatividad entre los dispositivos Bluetooth de distintos fabricantes, la especificación Bluetooth define un conjunto de perfiles. Cada uno de los perfiles definidos en [Bluetooth SIG01f] representa un posible escenario de uso en el que dos o más dispositivos dotados con tecnología Bluetooth deben interactuar para proporcionar al usuario un determinado servicio. En [Bluetooth SIG01f] se definen tanto los requisitos que un dispositivo debe satisfacer para implementar un determinado perfil, como los protocolos a utilizar y los procedimientos a seguir en distintos escenarios de aplicación. Entre los perfiles más destacados cabe mencionar:

- GAP (*Generic Access Profile*), que define procedimientos genéricos para la búsqueda de otras unidades y el establecimiento de conexiones.

- SDAP (*Service Discovery Application Profile*), que se centra en los mecanismos para el descubrimiento de servicios en otras unidades Bluetooth.
- PAN (*Personal Area Network*), que describe cómo dos o más unidades Bluetooth pueden formar una red *ad hoc* y cómo acceder a redes remotas a través de puntos de acceso utilizando los mismos procedimientos.
- SPP (*Serial Port Profile*), que define los requisitos de los dispositivos que quieran comunicarse mediante emulación de puerto serie.
- DUN (*Dial-Up Networking Profile*), que indica cómo proporcionar el servicio de acceso remoto a redes mediante la emulación de una conexión vía módem.
- LAN (*Local Area Network*) *Access Profile*, que especifica los mecanismos de interconexión entre dispositivos Bluetooth y redes de área local.³

En los apartados 2.4.1 y 2.4.2 se van a presentar los fundamentos de los perfiles PAN y SPP, que son los seleccionados para llevar a cabo el estudio de rendimiento en el capítulo 4. En dichos apartados 2.4.1 y 2.4.2 también se realiza una breve introducción a los protocolos RFCOMM y BNEP en los que se basan, respectivamente, PAN y SPP.

La motivación que ha propiciado la elección del perfil SPP ha sido, según se indicó en el capítulo 1, el hecho de que éste perfil es implementado por una amplia variedad de dispositivos, incluyendo la práctica totalidad de los dispositivos médicos comerciales, como los utilizados en los prototipos de sistemas de monitorización descritos en [Moron05a, Moron05b, Moron05c, Moron07a, Moron07c].

³El uso de este perfil ha sido reprobado en Bluetooth 1.2

Adicionalmente, se ha comparado el perfil SPP con el perfil PAN, ya que, aunque no es soportado en la actualidad por dispositivos comerciales, tiene la virtud de que posibilita la transmisión de tráfico IP sobre Bluetooth, gracias al uso del protocolo BNEP.

2.4.1. Perfil SPP

El Perfil de Puerto Serie define los protocolos y procedimientos a utilizar para la emulación de comunicación por cable serie RS232 (o similar).

En este perfil un dispositivo puede desempeñar dos papeles:

- Dispositivo A: Dispositivo que toma la iniciativa de establecer una conexión.
- Dispositivo B: Dispositivo que espera que otro dispositivo inicie el establecimiento de conexión.

Este perfil utiliza, como protocolo de transporte para la emulación de las comunicaciones por puerto serie, el protocolo RFCOMM, que es la adaptación para Bluetooth del protocolo GSM TS 07.10 [ETSI97]. RFCOMM se utiliza para transportar datos de usuario, señales de control de módem y comandos de configuración.

Con el fin de relacionar el perfil SPP con la arquitectura convencional de puerto serie, tanto el dispositivo A como el B pueden actuar como DCE (*Data Circuit-Terminating Equipment*) o como DTE (*Data Terminal Equipment*), ya que el protocolo RFCOMM está diseñado independientemente del sentido de interacción DTE-DCE o DCE-DTE.

2.4.1.1. Protocolo RFCOMM

RFCOMM es un protocolo de transporte simple que opera sobre L2CAP y que adicionalmente proporciona emulación de los 9 circuitos de puerto serie RS-232 (EIA/TIA-232-E) (*Electronic Industries Alliance / Telecommunications Industry Association*). El protocolo RFCOMM puede soportar hasta 60 conexiones simultáneas entre dos dispositivos Bluetooth.

No obstante, es conveniente aclarar que, en todo momento, sólo debe existir una sesión RFCOMM entre dos dispositivos, que es identificada por las direcciones BD_ADDR de ambos dispositivos. Así, cuando un dispositivo vaya a establecer una nueva DLC (*Data Link Connection*), debe verificar si ya hay una sesión RFCOMM establecida con ese dispositivo, de forma que en caso afirmativo, creará la nueva DLC sobre la sesión ya existente.

El protocolo RFCOMM está basado en el estándar ETSI TS 07.10 [ETSI97]. De hecho, se define el protocolo RFCOMM especificando un subconjunto del estándar ETSI TS 07.10 y algunas adaptaciones incluidas exclusivamente para Bluetooth. Concretamente, RFCOMM utiliza el modo de operación básico de TS 07.10. Los tipos de tramas TS 07.10 soportados por RFCOMM se especifican en la tabla 2.4.

Finalmente, cabe mencionar que RFCOMM es un protocolo que confía en que los niveles inferiores le proporcionen una transmisión fiable, por lo que cuando se agota el tiempo de espera de una respuesta del extremo remoto cierra la sesión.

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 2.4: Tipos de tramas TS 07.10 utilizadas por RFCOMM

2.4.2. Perfil PAN

El perfil PAN describe cómo dos o más dispositivos Bluetooth pueden formar una red *ad hoc*, así como los mecanismos a utilizar para acceder a una red remota a través de un punto de acceso a red.

Asimismo, la especificación del perfil PAN describe cómo usar el protocolo BNEP para proporcionar capacidades de interconexión de red a los dispositivos Bluetooth.

Los tres escenarios o casos de uso que contempla este perfil son los siguientes:

1. Punto de Acceso a Red o *Network Access Point (NAP)*: Un punto de acceso a red es una unidad que integra una o más radios Bluetooth y que actúa como un *bridge*, *proxy* o *router* entre una red Bluetooth y otra red de diferente tecnología (10BASE-T, GSM, etc).
2. Redes *ad hoc* o *Group ad hoc Network*: Permite a dispositivos Bluetooth crear, de forma cooperativa, redes inalámbricas *ad hoc* prescindiendo del uso de *hardware* o infraestructura adicional de interconexión de red. La versión 1.0 del perfil PAN se centra en el escenario de una red *ad hoc* personal

consistente en una única *piconet* Bluetooth con conexiones entre dos o más dispositivos.

3. Usuario de Red de Área Personal o *Personal Area Network User (PANU-PANU)*: En este escenario, una conexión punto a punto entre dos PANUs permite una comunicación directa exclusivamente entre esos dos nodos.

2.4.2.1. Protocolo BNEP (*Bluetooth Network Encapsulation Protocol*)

El protocolo BNEP encapsula paquetes de distintos protocolos de interconexión de redes, que serán transportados sobre el protocolo L2CAP. La especificación se basa en las siguientes asunciones:

1. El protocolo BNEP utiliza canales L2CAP orientados a conexión.
2. Bluetooth es considerado un medio de transmisión al mismo nivel OSI que Ethernet, Token Ring, ATM (*Asynchronous Transfer Mode*), etc.
3. L2CAP es considerado como la capa MAC de Bluetooth.
4. BNEP especifica una MTU (*Maximum Transfer Unit*) de L2CAP mínima de 1691 octetos.
5. Las reglas de conectividad de red y topología definidas en el estándar IEEE 802.3 (conmutación y enrutamiento) serán aplicadas a Bluetooth de forma consistente con el medio de comunicación IEEE 802.3.
6. El espacio de direcciones Bluetooth (BD_ADDR) es administrado por el IEEE, siendo asignado de acuerdo con el espacio de direcciones Ethernet. Esto significa que es posible disponer de un punto de acceso a red

Bluetooth que actúe como *bridge* entre dispositivos Bluetooth y una red Ethernet.

Capítulo 3

Revisión Bibliográfica:

Calidad de Servicio en Redes de Área Personal Bluetooth

El soporte de calidad de servicio es un aspecto crucial para garantizar la utilización del ancho de banda de las aplicaciones móviles basadas en redes WPAN Bluetooth [Chen05]. Teniendo en cuenta la amplia difusión experimentada por las WPAN, la transmisión de tráfico en tiempo real en una *piconet* Bluetooth, tal y como se apunta en [Ma05], es un tema de máximo interés en el área de investigación de las WPAN, que ha captado la atención de numerosos estudios científicos. En estos estudios el rendimiento se evalúa desde dos perspectivas:

- Considerando el efecto de la coexistencia con redes 802.11 (WiFi) o con otras *piconets* Bluetooth.

- En una *piconet* aislada, evaluando distintos parámetros relacionados con la calidad de servicio.

En los siguientes apartados se lleva a cabo una revisión del estado de la técnica en cada una de estas líneas de investigación.

3.1. Efecto de las interferencias en la calidad de servicio

En lo que concierne al impacto de las interferencias sobre el rendimiento se distinguen dos grandes líneas de investigación:

1. Coexistencia de *piconets* y redes IEEE 802.11 [Conti03, M. Cordeiro04, Corvaja06, Francia04, Golmie01a, Golmie01b, Golmie03a, Golmie03b, Golmie03c, Golmie04, Howitt03, Hsu06].
2. Coexistencia de distintas *piconets* [Ashraf06, Feng04, Li07a, Li07b, Pasolini03, Pasolini04, Peterson03, Yong05].

A continuación se realiza una breve revisión de los trabajos que se han considerado más representativos en cada uno de estos ámbitos.

3.1.1. Coexistencia Bluetooth - IEEE 802.11

De todas las contribuciones realizadas por Golmie, cabe destacar el trabajo presentado en [Golmie03b], en el que se evalúa el impacto de la interferencia

mutua entre Bluetooth y 802.11 sobre el rendimiento de sendas tecnologías. El análisis se realiza en un entorno de simulación desarrollado por los autores partiendo de modelos muy detallados de los niveles MAC y físico (PHY). Los autores parten de escenarios simples con el propósito de valorar parámetros como la potencia de transmisión, la carga y tipo de tráfico, para acabar con escenarios más complejos en los que coexisten múltiples *piconets* Bluetooth y dispositivos WLAN.

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

En [Conti03] se presenta un modelo analítico utilizado por los autores para llevar a cabo una evaluación del rendimiento de Bluetooth cuando es interferido por WiFi y viceversa. El modelo presentado contempla la existencia de ruido térmico, obstáculos en la propagación, interferencia, formatos de modulación, técnicas de codificación y, adicionalmente, la técnica de salto de frecuencia, las estructuras de los paquetes y distintas fuentes de tráfico Bluetooth. Finalmente, tras un exhaustivo análisis, para una determinada calidad de servicio definen el concepto de «dominio de coexistencia», expresado en función de distancias relativas.

En [Francia04], la coexistencia entre Bluetooth y 802.11 se analiza desde un punto de vista empírico. Se estudia el rendimiento de Bluetooth en términos de tasa de paquetes perdidos considerando varios escenarios, resultantes de combinar los modos de operación de 802.11: modo *ad hoc*, modo infraestructura

y ambos combinados, con Bluetooth. Los autores concluyen que, de las mediciones, se desprende que la tasa de paquetes perdidos es mayor en modo *ad hoc* que 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 expuesto en [M. Cordeiro04] se caracteriza también por un enfoque práctico. Se describe una arquitectura, que los autores presentan 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 WLAN sin necesidad de utilizar puntos de acceso Bluetooth fijos. Para minimizar el efecto de la interferencia entre Bluetooth y 802.11, combinan las técnicas AFH y de detección de portadora Bluetooth (BCS, *Bluetooth Carrier Sense*). Adicionalmente, los autores obtienen, matemáticamente, una cota superior del número de BWG requeridos en una *scatternet* Bluetooth para garantizar acceso ininterrumpido a la red inalámbrica.

En [Hsu06] los autores presentan un método de coexistencia entre WiFi y Bluetooth no cooperativo consistente en optimizar dinámicamente, en función de la tasa de paquetes erróneos (PER, *Packet Error Rate*), la longitud del paquete de la capa MAC 802.11 para que el dispositivo WiFi tenga más posibilidades de evitar la interferencia causada por Bluetooth. Los autores concluyen que, aunque la propuesta no supone una mejora apreciable para las prestaciones de las comunicaciones Bluetooth, se consigue un incremento significativo en el rendimiento de WiFi, en términos de *throughput* y de retardo de transmisión.

En [Corvaja06] se analiza el rendimiento de una red Bluetooth-802.11b. El objetivo de los autores es maximizar la calidad de servicio permitiendo a los usuarios conmutar de una red a otra. Para ello, se evalúan parámetros de calidad de servicio, tales como la latencia (o retraso) de paquete, la tasa de paquetes erróneos y el *throughput*, en presencia de un procedimiento de *handover* vertical y teniendo en cuenta la interferencia mutua Bluetooth-WiFi.

3.1.2. Coexistencia entre *piconets*

En [Pasolini03] y [Pasolini04] se realiza un estudio analítico sobre el *throughput* en un escenario en el que coexisten múltiples *piconets*. El objetivo del autor es encontrar una cota inferior del *throughput* total y determinar el número de *piconets* que maximiza esa cota.

En [Feng04] los autores explican que, aunque tradicionalmente las investigaciones en el contexto de la interferencia entre *piconets* se centran en torno a la probabilidad de colisión, no se tiene en cuenta que esta probabilidad disminuye con la distancia. Considerando este hecho, determinan la tasa de paquetes erróneos (PER) en función de la distancia, indicando también el tamaño de paquete óptimo en presencia de un número variable de *piconets*.

En relación con el cálculo de la probabilidad de colisión, en [Peterson03] se calcula una cota superior de la tasa de paquetes erróneos para paquetes de 1 *slot* teniendo en cuenta dos factores de dependencia: (1) las frecuencias de la secuencia de salto son elegidas de un conjunto de 79 frecuencias, que depende de las ya usadas; (2) las frecuencias de paquetes consecutivos son seleccionadas de conjuntos mutuamente excluyentes. Los autores concluyen que la cota resultante

es un 24 % superior a la obtenida cuando se asume independencia en la secuencia de salto.

En el mismo ámbito, los autores de [Ashraf06] van más allá proponiendo un esquema de sincronización en el tiempo entre las *piconets* coexistentes para optimizar el rendimiento de conexiones ACL. El método propuesto, que se centra en el uso de paquetes DH, conlleva la introducción de un estado adicional en el nivel BB. Esta técnica es evaluada analíticamente en términos de tasa de paquetes erróneos (PER), *throughput* de cada *piconet* y *throughput* total, y los autores concluyen que permite mejorar en un 26 % el *throughput* total en un escenario de 100 *piconets* respecto al caso en el que no se sincronizan las transmisiones de las distintas *piconets*.

Los trabajos presentados en [Li07a, Li07b] se centran en reducir la tasa de paquetes erróneos (PER) producida por colisiones entre las transmisiones de múltiples *piconets*. Con este fin, los autores proponen un esquema de transmisión redundante, denominado DCT (*Dual Channel Transmission*), consistente en enviar el mismo paquete por dos canales distintos, de forma que la potencia utilizada por cada canal sea la mitad de la que se emplearía en la transmisión por un único canal (*SCT: Single Channel Transmission*). Los autores concluyen que cuando el número de *piconets* es menor que 20 y la SNR (*Signal to Noise Ratio*) es superior a 18 dB (15 dB en cada canal DCT), el esquema DCT permite lograr una mejora significativa en términos de PER, *throughput* y retardo medio de transmisión respecto al esquema de transmisión convencional SCT.

El estudio del rendimiento en una *piconet* Bluetooth, expuesto en [Yong05], se lleva a cabo considerando, primero, la interferencia provocada por una única

piconet y, posteriormente, extendiéndose para el caso de que existan múltiples *piconets*. Como medida de rendimiento, se utiliza la tasa PER, obtenida a partir de la BER (*Bit Error Rate*), definida en función de la relación SNR instantánea. El desarrollo analítico se realiza para paquetes de un *slot* y asumiendo que el tráfico de cada *piconet* consume todos los *slots*. Para una única *piconet* interferente, el cálculo de probabilidad se efectúa, considerando la probabilidad de colisión en frecuencia y distinguiendo varios casos, atendiendo a las distintas partes del paquete a las que puede afectar la interferencia. En el segundo escenario (tres *piconets* interferentes) se distinguen tres casos, dependiendo de que las tres *piconets*, dos, o ninguna coincidan en frecuencia. Finalmente, los autores comparan los resultados calculados analíticamente con los obtenidos por simulación mediante la herramienta OPNET [Opnet], concluyendo que éstos prueban el análisis numérico realizado.

En resumen, los trabajos de investigación que abordan el estudio de la coexistencia entre *piconets* o de éstas con 802.11, cubren el análisis de un amplio espectro de factores a tener en cuenta: la probabilidad de colisión en tiempo o frecuencia, y su dependencia de la distancia, la utilización de técnicas de sincronización entre las diferentes tecnologías para minimizar la interferencia o, incluso, el uso de esquemas de transmisión redundantes. Sin embargo, no se ha encontrado que se evalúe, experimentalmente, el efecto de la interferencia sobre el retardo en transmisiones Bluetooth. Además, indistintamente del tema que sea objeto de análisis en el ámbito de la coexistencia, en la mayoría de los casos, estos estudios se avalan mediante simulaciones, recurriéndose a la validación empírica en un número limitado de ocasiones, como es el caso de los estudios ya mencionados [Francia04] y [M. Cordeiro04].

3.2. Calidad de servicio en *piconets*

Aunque el tráfico de datos se proporciona como un servicio *Best Effort* sobre conexiones ACL, existe un considerable número de propuestas para optimizar la eficiencia de las transmisiones Bluetooth [Ma05]. En esta área predominan los estudios que versan sobre el esquema de *polling* utilizado por el dispositivo Bluetooth que actúa como maestro [Jiang05, Ma05, Yen05]. Objeto de investigación es también el estudio del *throughput* y del retardo extremo a extremo en función de la distancia entre el origen y destino [Francia04], o la tasa BER [Huang05]. Además, existe un gran número de aportaciones cuyo objetivo es el estudio del tráfico TCP (*Transmission Control Protocol*) sobre redes Bluetooth, evaluado para distintas implementaciones de TCP [Chen04, Das01], diferentes tipos de segmentación y diversos esquemas de *polling* [Chan04, Das01].

Concretamente, el estudio presentado en [Chen04] se centra en 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 el *throughput* de TCP puede mejorarse significativamente si se utiliza un paquete de tamaño determinado.

En [Das01] se estudian dos políticas de segmentación y ensamblado (SAR) —SAR-*Best Fit* (BF) y SAR-*Optimum Slot Utilization* (OSU)—, concluyéndose que es el método SAR-OSU el que permite lograr un mayor rendimiento. Utilizando este método, los mismos autores continúan evaluando el efecto del tamaño de los *buffers* de L2CAP, de los algoritmos de planificación y de las distintas variantes de las implementaciones de TCP.

Cuando se analiza el tráfico TCP, normalmente, no se contempla el uso de un perfil específico para el transporte de datos TCP/IP (*Internet Protocol*), sino que se encapsula directamente en PDUs (*Protocol Data Unit*) de datos de L2CAP. Es el caso de los trabajos presentados en [Das01] y [Bell03] que, con el objetivo de minimizar la sobrecarga (*overhead*), justifican la no utilización de perfiles. La razón que argumentan es que el uso del perfil DUN supone una sobrecarga bastante elevada, ya que, a la introducida por las cabeceras de TCP/IP o UDP/IP, se añade la correspondiente a los protocolos PPP (*Point-to-Point Protocol*) y RFCOMM. En [Bell03] se descarta el empleo del perfil PAN por el mismo motivo, mientras que en [Das01] no se considera esta posibilidad porque en el momento de la elaboración del trabajo, no estaba disponible la especificación de dicho perfil.

Una consecuencia del envío directo de datos sobre L2CAP es que en los estudios realizados, como el presentado en [Das01], sólo se tiene en cuenta el efecto sobre el rendimiento (*throughput* y retardo extremo a extremo) de la fragmentación de datos de usuario llevada a cabo por la unidad SAR del nivel L2CAP para su transmisión en paquetes Bluetooth, y no de la que se produce cuando se excede la MTU del nivel inferior.

Por otro lado, en los estudios en los que se considera el uso del protocolo BNEP [Ali05, Chan04], específicamente pensado para el transporte de datos IP sobre Bluetooth, no se lleva a cabo un estudio equivalente al realizado para RFCOMM en [Okura02]. Este último estudio contempla el efecto del «desajuste por segmentación» —*segmentation mismatch*— que puede producirse cuando no hay coincidencia en el máximo tamaño de los datos manejable entre capas adyacentes de la arquitectura. En el trabajo presentado en [Chan04], por contra,

se tiene en cuenta la forma en la que el tráfico TCP se transporta sobre BNEP, configurando el parámetro MSS (*Maximum Segment Size*) a un valor elevado para conseguir el menor retardo de paquete extremo a extremo. También en [Ali05] se valora el *overhead* que supone la cabecera del datagrama IP para transmisiones de voz sobre IP utilizando BNEP y se propone reducir el retardo en dichas transmisiones comprimiendo la cabecera IP. Sin embargo, en [Latkoski07], donde se realiza un análisis del tráfico Web generado desde PDAs o portátiles a través de una conexión Bluetooth con un teléfono móvil 3G que actúa de pasarela a Internet, no se considera el *overhead* añadido a los datos de usuario por cada uno de los protocolos que intervienen en la comunicación.

En lo que respecta al estudio de la gestión de calidad, hay que resaltar que en [Wu04] los autores comienzan con una reflexión acertada sobre el enfoque práctico del estudio de métodos orientados a mejorar la calidad de servicio en redes Bluetooth: Bluetooth es una tecnología inalámbrica prometedora para redes de área personal y está siendo aplicada para múltiples servicios, basados o no en IP. Sin embargo, el estándar Bluetooth no especifica cómo cumplir los requisitos de calidad y aunque existen múltiples propuestas, los enfoques planteados suponen la modificación de la especificación Bluetooth y de los dispositivos actuales. Bajo esta premisa, los autores presentan en [Wu04] un conformador de tráfico a nivel de L2CAP y evalúan, utilizando la pila de protocolos BlueZ¹ [BlueZ06] y dispositivos reales, el rendimiento para distintos perfiles Bluetooth.

Al margen del funcionamiento de dispositivos reales y en relación con el

¹Véase el capítulo 5, donde se realiza una breve introducción a BlueZ, implementación de la pila de protocolos Bluetooth proporcionada por Linux.

análisis teórico de los esquemas de *polling*, se encuentra el trabajo presentado en [Jiang05]. En éste los autores recuerdan que con el algoritmo de planificación convencional RR (*Round Robin*), el orden en el que los esclavos son direccionados es fijo, independientemente de que tengan datos que transmitir o no, provocando, por tanto, un desaprovechamiento del ancho de banda. Para evitarlo proponen alterar dinámicamente este orden, teniendo en cuenta el tráfico generado por cada esclavo y utilizando un planificador MAC basado en prioridad para transmitir tráfico en tiempo real sobre conexiones de ACL con soporte de QoS. La comparación de la mejora conseguida en el rendimiento la llevan a cabo mediante simulación software, utilizando la herramienta OPNET [Opnet]. En [Yen05] también se busca una alternativa a la política de planificación RR para las transmisiones de un maestro a diferentes esclavos, comparándose, mediante simulación, dicho algoritmo con el ERR (*Exhaustive Round Robin*) y con el propuesto por los autores: RR-FCFS (*Round Robin-First Come First Serve*). En la misma línea de investigación se centran los trabajos presentados en [Ma05] y [Mistic04]. En [Ma05] los autores proponen un esquema de *polling* adaptativo para soportar servicios de tiempo real en una WPAN Bluetooth y evalúan la relación entre el retardo y la carga de tráfico, también mediante simulación. En [Mistic04] se expone un esquema de *polling*, también adaptativo, que los autores denominan ACLS (*Adaptative Cycle Limited Scheduling*), comprobando por simulación que permite lograr un rendimiento prácticamente óptimo para tráfico asíncrono así como garantizar una determinada calidad para tráfico CBR (*Constant Bit Rate*).

Adicionalmente, cabe destacar la aportación del estudio [Chan04] ya citado, por el hecho de que se proponen esquemas de *polling* orientados a mejorar el

rendimiento del tráfico TCP sobre Bluetooth. Esto es, los autores no se limitan a optimizar la eficiencia del tráfico Bluetooth en el nivel BB, como se hace por regla general, sino que también tienen en cuenta la forma en la que ese tráfico se transporta sobre BNEP, como ya se ha explicado anteriormente. En estos casos, la evaluación de la política de gestión de calidad se lleva a cabo mediante simulación y no sobre comunicaciones entre dispositivos reales.

Entre los estudios científicos que contemplen el efecto de las retransmisiones en los enlaces ACL cabe mencionar los presentados en [Madhavapeddy05], [Chen04], [Khoutaif06] y [Huang05].

Los trabajos [Madhavapeddy05] y [Chen04] utilizan la tasa de bits erróneos (BER), proporcionada indirectamente por el dispositivo e interpretada de acuerdo con las especificaciones del propio fabricante [CSR]. No obstante, sus pretensiones son muy diferentes. El trabajo [Madhavapeddy05] presenta un sistema de localización basado en la calidad de los enlaces Bluetooth, así como el experimento llevado a cabo para evaluarlo. Sin embargo, el objetivo de los autores de [Chen04], como ya se ha indicado anteriormente, es estudiar el efecto de la variabilidad de la calidad del enlace en el *throughput* de transmisiones TCP sobre Bluetooth, con el fin de adaptar el tamaño del paquete óptimo a las condiciones del canal. Con este fin, en primer lugar, calculan el *throughput* en función de la tasa BER, y deducen el tipo de paquete óptimo para varios intervalos de la tasa BER, seleccionando siempre aquel que maximice el *throughput*. En una segunda fase, presentan los resultados de la simulaciones realizadas para comparar la ventaja de adaptar el tipo de paquete. Las simulaciones fueron ejecutadas con un modelo que desarrollaron como una extensión del NS-2 (*Network Simulator*) [NS-2] con tasa BER fija, variable

y medida. En el último caso, la metodología consistió en introducir los valores de BER medidos con dispositivos reales en presencia de una red 802.11.

En [Huang05] se presenta un modelo teórico del retardo para la transmisión de paquetes ACL, definiendo el retardo como el intervalo de tiempo que transcurre desde que un paquete ACL se almacena en el *buffer* del nivel BB hasta el instante en el que se recibe una confirmación positiva. En el desarrollo analítico se tiene en cuenta el número medio de retransmisiones realizadas, expresado en función de la tasa de paquetes erróneos, así como los enlaces SCO establecidos, el número de esclavos activos y los estados en los que se encuentran las conexiones de la *piconet*. Para presentar los resultados del modelo expuesto, los autores representan el retardo de transmisión de los paquetes DH y DM en función del número de esclavos en estado activo, para distinto número de conexiones en modo *PARK* y diferentes valores de tasa BER. Sin embargo, el trabajo no define la relación entre la BER y la tasa de paquetes erróneos utilizada para calcular el número medio de retransmisiones.

En [Khoutaif06] los autores describen un modelo formal basado en redes de Petri, realizando previamente un análisis exhaustivo de las probabilidades de detección de errores para la cabecera y el campo de datos de los paquetes Bluetooth. El estudio contempla que los errores puedan afectar a determinados bits, como el bit SEQN de la cabecera, o alguno de los 16 bits del CRC del campo de datos. Aunque no presentan el método de verificación del modelo descrito, su conclusión se centra en proponer que el esquema de confirmación de paquetes utilizado por el nivel BB de Bluetooth debería ser numerado. Como justificación, describen el caso en el que se acaba descartando un paquete porque, debido

a un error en el bit SEQN, se interpreta que dicho paquete corresponde a una retransmisión innecesaria de una paquete previo correctamente recibido.

En resumen, en la mayor parte de los trabajos de investigación se obvia el efecto del *overhead* añadido por cada uno de los protocolos que intervienen en los perfiles que puede implementar un dispositivo comercial, así como el impacto de la fragmentación que se puede producir cuando no existe coincidencia entre las máximas longitudes de los datos manejados en capas adyacentes de la arquitectura de protocolos. Además, salvando el trabajo [Wu04], anteriormente citado, se echa en falta la realización de estudios empíricos del retardo usando perfiles, que evalúen la repercusión de la gestión de calidad a través de los medios proporcionados por la especificación Bluetooth, en especial la proyección de los parámetros de calidad sobre el mecanismo de *polling*.

La importancia de tener en cuenta el uso de perfiles radica en el hecho de que prescindir de su uso no es una opción factible en sistemas en los que se integran dispositivos reales que implementan un determinado perfil y sobre los que se dispone de un control nulo o muy escaso.

Por estos motivos, el objetivo del presente trabajo es realizar un estudio del rendimiento en una red de área personal WPAN basada en los perfiles Bluetooth SPP y PAN. El rendimiento para los perfiles SPP y PAN se va a evaluar en términos de retardo extremo a extremo y *throughput*, teniendo en cuenta la influencia sobre el retardo del *overhead* introducido a lo largo de toda la arquitectura Bluetooth y el efecto del *segmentation mismatch*. El estudio consiste en proponer un modelo analítico de retardo que se contrastará con medidas experimentales. Adicionalmente, para el caso de SPP, perfil soportado por una

amplia mayoría de dispositivos comerciales, se extiende el modelo con el fin de evaluar cómo afecta la configuración de calidad de servicio y la existencia de retransmisiones al retardo extremo a extremo.

En este punto, es conveniente aclarar que el objetivo que se pretende lograr con esta tesis, al evaluar cómo afecta la ocurrencia de retransmisiones al retardo, es completar el modelo analítico que se desarrollará para el caso sin pérdidas. La extensión del modelo contemplará los efectos de las retransmisiones, pero sin entrar a valorar la naturaleza de la interferencia que provoca las pérdidas. Para ello, se derivará el número medio de retransmisiones a partir de la tasa PER ya que, tal y como se indica en [Yong05], es el parámetro que se utiliza como medida de rendimiento en entornos ruidosos. Esta tasa PER, a su vez, se expresará en función de la tasa de bits erróneos (BER). Finalmente, la BER estimada por el dispositivo se introducirá en el modelo teórico, para contrastarlo con el retardo de transmisión medido en escenarios de transmisión real.

Capítulo 4

Estudio teórico del rendimiento en *piconets* Bluetooth utilizando perfiles

En este capítulo se expone un estudio teórico del rendimiento que se centra en la elaboración de un modelo construido progresivamente. El objetivo inicial es determinar analíticamente el retardo extremo a extremo y el *throughput* en condiciones óptimas. En segunda instancia, se modifica el modelo teniendo en cuenta la gestión de la calidad de servicio prevista en la especificación Bluetooth [Bluetooth SIG01c] y, finalmente, se extiende el análisis teórico para incorporar la existencia de retransmisiones, que tienen lugar en la práctica por errores de transmisión o como consecuencia de la coexistencia con otros sistemas o tecnologías que operan en la banda ISM.

En concreto, en la sección 4.1, se comienza estimando el retardo mínimo, considerando que los datos de usuario a transmitir se entregan directamente al nivel BB (apartado 4.1.2). A continuación, partiendo de este retardo mínimo, en los apartados 4.1.3 y 4.1.4, para los perfiles PAN y SPP, se determina el retardo teniendo en cuenta el *overhead* añadido por cada uno de los protocolos involucrados en cada caso, así como la diferencia que puede existir entre el tamaño de las unidades de transferencia de datos manejadas en niveles adyacentes. Adicionalmente, en el apartado 4.1.6, se determina el *throughput*, teniendo en cuenta el modelo de retardo presentado en los apartados anteriores.

La sección 4.1 también incluye, en el apartado 4.1.5, un análisis comparativo del rendimiento de ambos perfiles (SPP y PAN) y una discusión sobre las ventajas e inconvenientes del empleo de cada uno de ellos desde el punto de vista del rendimiento (apartado 4.1.7).

Las secciones 4.2 y 4.3 se centran en el caso de SPP y proponen, respectivamente, cómo modificar el modelo de retardo propuesto en la sección 4.1 para introducir la repercusión de la configuración de calidad de servicio y la existencia de retransmisiones.

4.1. Perfiles SPP y PAN: Efecto del *overhead* y del *segmentation mismatch* en el retardo

4.1.1. Introducción

La justificación que se argumenta para centrar este estudio en los perfiles PAN y SPP es que, para una WPAN Bluetooth, el perfil PAN podría resultar la opción más apropiada, sobre todo si se pretende:

- Que los dispositivos que integran la WPAN puedan ser direccionados remotamente de forma independiente.
- Que el coordinador de la PAN, que soportaría el servicio NAP, actúe como un simple encaminador de paquetes, de forma que el tráfico generado por cada uno de los usuarios de la PAN (*PAN Users*), sea encaminado hacia otra red diferente, por ejemplo, la red móvil.
- Proporcionar un servicio basado en IP.

Estas características serán ventajas o inconvenientes, dependiendo de los requisitos de la aplicación concreta. Por ejemplo, el perfil PAN puede resultar el más adecuado para aplicaciones basadas en redes *ad hoc*, pero no para un sistema de monitorización médica basado en una red de área personal o corporal (BAN, *Body Area Network*). En el primer caso es imprescindible que cada nodo pueda direccionar a los demás, mientras que en el segundo caso suele existir un nodo concentrador que actúa de interfaz entre los dispositivos de la PAN o de la BAN y otras redes. En este tipo de aplicaciones puede preferirse el uso de otros perfiles, como el de SPP frente a PAN. De hecho, para

el perfil SPP, implementado por una gran variedad de dispositivos comerciales, podría lograrse un mayor rendimiento, ya que el *overhead* introducido es considerablemente inferior al que se añade a los datos para su transporte en TCP/IP (o UDP/IP) sobre BNEP.

Con el fin de verificar esta hipótesis, se comienza en el apartado 4.1.2 presentando una estimación teórica del retardo de transmisión mínimo, suponiendo que los datos a enviar se entregan directamente al nivel BB. A continuación, en el apartado 4.1.3 se considera el *overhead* añadido por cada uno de los protocolos que intervienen en la transmisión al utilizar el perfil SPP. Posteriormente, en el apartado 4.1.4 se repite el estudio para el perfil PAN, exponiéndose con más detalle que en los trabajos [Moron07b] y [Moron08], y en último lugar, se compara el retardo mínimo con el obtenido analíticamente para cada uno de estos perfiles.

La figura 4.1 muestra, sobre la arquitectura de protocolos Bluetooth, el retardo de transmisión que se va a calcular para cada uno de los perfiles: t_R en el caso de SPP y t_U en el caso de PAN, a partir del retardo mínimo, t_{TX} , calculado en 4.1.2.

En todos los casos la estimación teórica del retardo de transmisión se efectúa asumiendo condiciones ideales, es decir, que no se realizan retransmisiones a nivel BB y que el tiempo de almacenamiento en los *buffers* es despreciable.

4.1.2. Cálculo del retardo mínimo de transmisión

Como ya se ha indicado, en este apartado se lleva a cabo una estimación teórica del retardo de transmisión mínimo considerando que los datos de

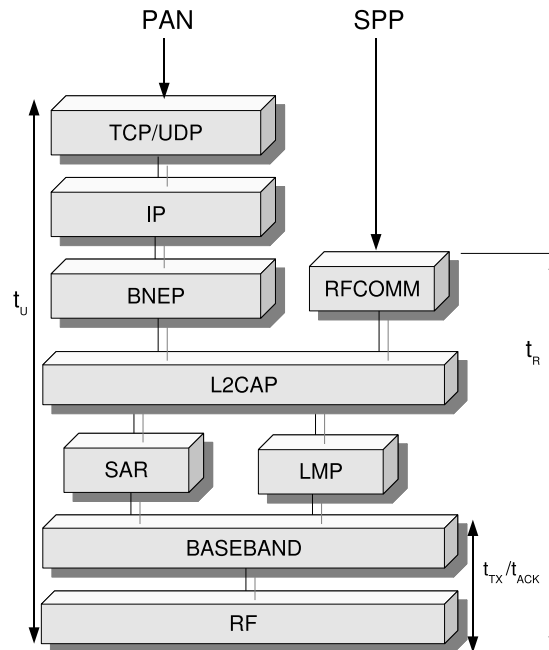


Figura 4.1: Arquitectura de protocolos Bluetooth

usuario a enviar por Bluetooth se entregan directamente al nivel BB, esto es, obviando las capas de nivel superior.

El retardo mínimo de transmisión de N octetos de usuario, según el esquema TDD de transmisión en una *piconet* (véase el apartado 2.1.2.1), tiene lugar en el caso de que el flujo de información a transmitir vaya del maestro al esclavo, asumiendo que la confirmación de cada paquete Bluetooth se realiza en el siguiente *slot* y con un paquete de un *slot*.

Para contemplar el hecho de que el número de octetos de usuario pueda exceder la máxima capacidad de transporte de un paquete DH5, es necesario definir dos componentes del retardo, $t_{ACK}(N)$ y $t_{TX}(N)$, que se describen a continuación:

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

$$t_{ACK}(N) = \begin{cases} 0 & N = 0 \\ 2 \cdot T_S & 0 < N \leq L_1 \\ 4 \cdot T_S & L_1 < N \leq L_3 \\ 6 \cdot T_S & L_3 < N \leq L_5 \\ 6 \cdot T_S \cdot \left\lfloor \frac{N}{L_5} \right\rfloor + & N > L_5 \\ + t_{ACK} \cdot (N \bmod L_5) & \end{cases} \quad (4.1)$$

donde el operador $\lfloor Y \rfloor$ redondea el número Y al entero inferior.

En la fórmula anterior, T_S denota la duración de un *slot* Bluetooth ($625 \mu s$), y 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, 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.

Nótese que la ecuación 4.1 computa el *slot* de confirmación de la transmisión de un paquete que transporte un fragmento intermedio, ya que el maestro no transmitirá el siguiente fragmento hasta no haber recibido la confirmación del actual. Por consiguiente, 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*.

- $t_{TX}(N)$: Representa el tiempo empleado en la transmisión de N 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, 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:

$$t_{TX}(N) = \begin{cases} 0 & N = 0 \\ n_b(N) \cdot t_b & 0 < N \leq L_5 \\ t_{ACK}(L_5) \cdot \left\lfloor \frac{N}{L_5} \right\rfloor + & N > L_5 \\ +t_{TX}(N \bmod L_5) & \end{cases} \quad (4.2)$$

donde:

- t_b es el tiempo de transmisión de un bit, $1 \mu s$ (teniendo en cuenta que Bluetooth transmite a una velocidad de 1 Mbit/s).
- $n_b(N)$ 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:

$$n_b(N) = n_d(N) + C_{BB} \quad (4.3)$$

donde:

- C_{BB} es el tamaño del *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.

- $n_d(N)$ 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 2.5):

$$n_d(N) = \begin{cases} (N + C_{df}(N) + 2) \cdot 8 & \text{paquetes DH} \\ \left\lceil \frac{(N + C_{df}(N) + 2) \cdot 8}{10} \right\rceil \cdot 15 & \text{paquetes DM} \end{cases} \quad (4.4)$$

donde el operador $\lceil Y \rceil$ redondea el número Y al entero superior. En la fórmula anterior, el tercer sumando (2 octetos) contabiliza el *overhead* asociado al campo CRC, mientras que $C_{df}(N)$ representa el número de octetos correspondientes a la cabecera del *payload*, de acuerdo con la expresión:

$$C_{df}(N) = \begin{cases} 1 & N \leq L_1 \\ 2 & L_1 < N \leq L_5 \end{cases} \quad (4.5)$$

La ecuación 4.4 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.

Obsérvese que en la ecuación 4.2:

- Si N es menor o igual que L_5 , 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 y, tras el procesamiento que proceda realizar, los entrega al *Host*.

- Si N es mayor que L_5 , no se considera el tiempo de confirmación del último paquete transmitido, pero sí el de los fragmentos intermedios. De ahí que se invoque la fórmula de $t_{ACK}(N)$.

En este punto cabe señalar que en la definición del retardo de transmisión asociado a cada uno de los perfiles, $t_R(N)$ y $t_U(N)$, que se realizará en los siguientes apartados, ha de tenerse en cuenta el número de octetos de las cabeceras añadidas por cada uno de los niveles subyacentes, así como la necesidad de tener que fragmentar en cada nivel para no exceder la MTU impuesta por el nivel inferior. Para ello, será necesario hacer uso de la definición de las componentes $t_{ACK}(N)$ y $t_{TX}(N)$, que tienen en cuenta la fragmentación realizada a nivel BB de un mensaje del nivel superior.

4.1.3. Cálculo del retardo para el perfil SPP

Como ya se ha dicho anteriormente, el protocolo RFCOMM, que es un subconjunto del estándar ETSI TS 07.10 [ETSI97], es utilizado por el perfil SPP para permitir la emulación de puertos serie sobre L2CAP.

La estructura de la trama utilizada por RFCOMM, que se muestra en la figura 4.2, modifica la definida en TS 07.10, excluyendo el campo FLAG del inicio y final de la trama. La codificación del campo CONTROL, que indica la trama a utilizar, se especifica en la tabla 2 de la norma ETSI TS 07.10 [ETSI97].

La trama en la que RFCOMM transmite los datos de usuario es de tipo UIH (véase tabla 2.4 del apartado 2.4.1.1). Esta trama, en el campo de información (*Information*), puede transportar un número de octetos que viene impuesto por el parámetro $N1$ (*Maximum Frame Size*), cuyo valor por defecto es 127 octetos, pero que las entidades homólogas pueden negociar entre 23 y 32767 octetos, mediante el par comando/respuesta PN (*Parameter Negotiation*).

En las tramas UIH con el bit P/F (existente en el campo de control) a cero, el máximo tamaño del campo de información es el negociado, $N1$, mientras que cuando se está utilizando el control de flujo basado en créditos (bit P/F a uno), la estructura de la trama UIH se ve alterada, de forma que el máximo número de octetos, $N1$, es decrementado en uno para compensar el campo de créditos (manteniendo así el tamaño del *payload* del L2CAP constante).

Puesto que en el estudio realizado no se considera esta última modificación, para transmitir N octetos mediante RFCOMM, el efecto de la fragmentación y el *overhead* introducidos sobre el retardo de transmisión, $t_R(N)$, vendrá dado por la ecuación 4.6:

$$t_R(N) = n_{fnf_R} \cdot t_{ACK}(L_R + O_R(L_R) + C_L) + \quad (4.6)$$

$$+ t_{TX}(L_{ff_R} + O_R(L_{ff_R}) + C_L)$$

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 4.2: Estructura de la trama del Modo de Operación Básico del protocolo TS 07.10 (fuente: parte F:1 de la especificación [Bluetooth SIG01c])

siendo:

- L_R : longitud a la que RFCOMM tendrá que 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 $N1$.

Así, RFCOMM generará fragmentos de tamaño:

$$L_R = \min(N1, M_R - 5) \quad (4.7)$$

En la fórmula anterior, si la suma del parámetro $N1$ negociado con el nivel homólogo y de la sobrecarga máxima de RFCOMM (5 octetos) excede M_R , en lugar de fragmentar los datos de acuerdo con el valor del parámetro $N1$ negociado, tendría que hacerlo en bloques de un tamaño impuesto por la MTU de L2CAP¹ (véase la figura 4.3).

- $O_R(L)$: número de octetos correspondientes 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 (véase la figura 4.2):

$$O_R(L) = \begin{cases} 4 & 0 < L \leq 127 \\ 5 & L > 127 \end{cases} \quad (4.8)$$

- C_L : 4 octetos correspondientes al tamaño de la cabecera de la PDU de L2CAP en Bluetooth 1.1, equivalente a la PDU de L2CAP del modo básico del protocolo L2CAP en Bluetooth 1.2 (véase la figura 4.4).

¹Es responsabilidad del protocolo del nivel superior limitar el tamaño de los paquetes enviados al nivel L2CAP por debajo del límite de la MTU (parte D de la especificación [Bluetooth SIG01c]).

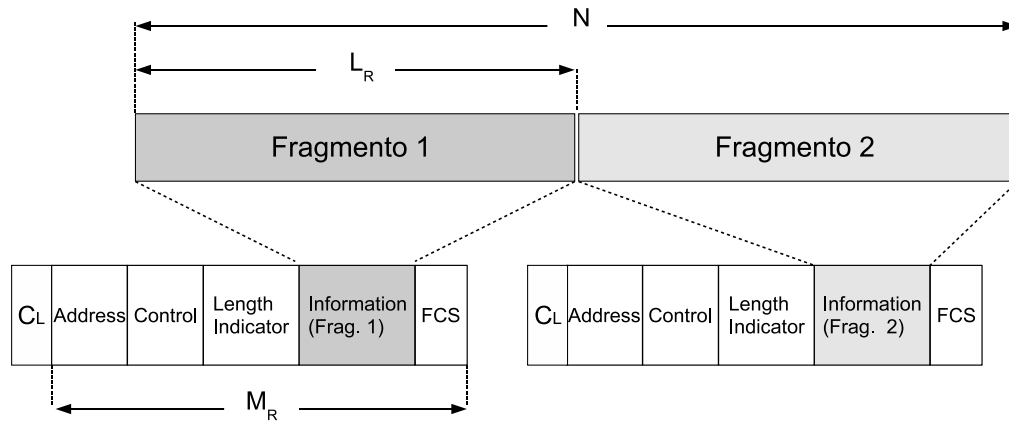


Figura 4.3: Fragmentación realizada en el nivel RFCOMM cuando el tamaño de los datos de usuario excede el valor de L_R , dado por la ecuación 4.7

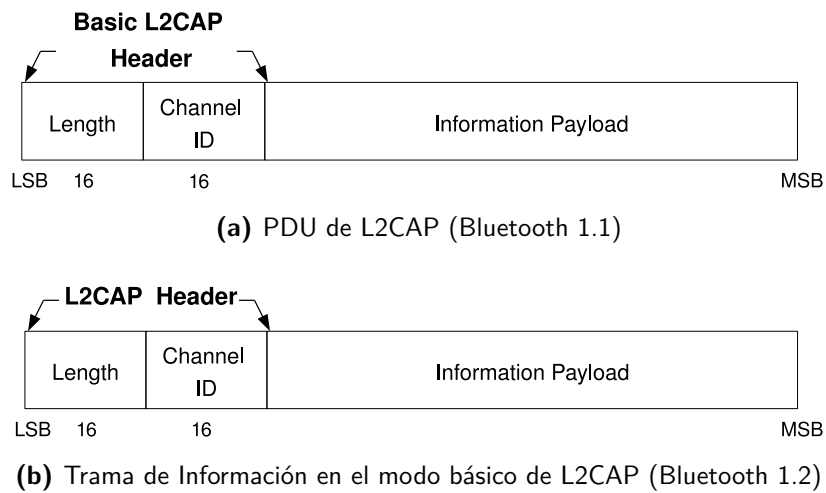


Figura 4.4: PDU de L2CAP en Bluetooth 1.1 (fuente: parte D de la especificación [Bluetooth SIG01c]) y Bluetooth 1.2 (fuente: parte A del volumen 3 de la especificación [Bluetooth SIG03a])

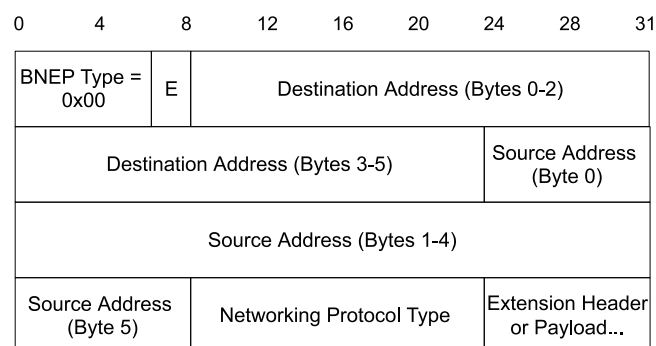
- $n_{fnf_R} = \left\lceil \frac{N}{L_R} \right\rceil - 1$: número de fragmentos no finales, teniendo en cuenta que los N octetos de datos se dividen en fragmentos de L_R octetos.
- $L_{ff_R} = ((N - 1) \bmod L_R) + 1$: número de octetos correspondientes al último fragmento. Nótese que, si los datos caben en un único fragmento ($N \leq L_R$), se tendrá que $n_{fnf_R} = 0$ y $L_{ff_R} = N$.

La especificación Bluetooth no establece una MTU de L2CAP mínima para RFCOMM, esto es, M_R no tiene un valor predeterminado o recomendado por la norma como en el caso de BNEP, que se verá en el apartado 4.1.4. De este modo, si no se fija el valor de M_R de acuerdo con el parámetro $N1$, en el caso de que $N1$ supere el valor de M_R , RFCOMM tendrá que fragmentar los datos de usuario ajustándose a la MTU de L2CAP (inferior a $N1$). En consecuencia, el *overhead* transmitido será mayor que en el caso de que exista concordancia entre el parámetro $N1$ y la MTU M_R , porque se requerirán más tramas RFCOMM para el envío de una determinada cantidad de datos de usuario. Por ejemplo, si $N1$ y M_R se fijasen respectivamente a 1019 y 128 octetos, la transmisión de 1500 octetos de datos de usuario se tendría que hacer en 13 tramas RFCOMM (es decir, 12 de 124 octetos y una última de 12 octetos), lo que implicaría un *overhead* de 104 octetos ($13 \cdot (O_R(124) + C_L)$). Sin embargo, si para M_R se elige un valor de 1024 octetos, tan sólo se precisarían 2 tramas RFCOMM, de 1019 y 481 octetos, con un *overhead* de 10 octetos.

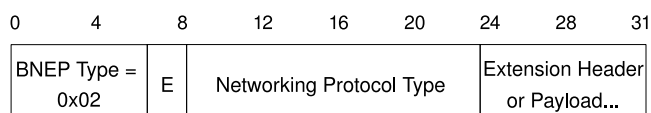
4.1.4. Cálculo del retardo para el perfil PAN

Como ya se ha apuntado, el perfil PAN permite el transporte de datos TCP/IP (o UDP/IP) sobre L2CAP mediante el protocolo BNEP. Para

ello, BNEP reemplaza la cabecera Ethernet (típica de las conexiones LAN cableadas) por su propia cabecera, de forma que la cabecera BNEP y el *payload* de Ethernet serán encapsulados en una PDU de datos de L2CAP. A la información transportada por la trama Ethernet se añadirían un total de 15 octetos, correspondientes a la cabecera BNEP, si se emplea un paquete del tipo general (*BNEP_GENERAL_ETHERNET*), cuya estructura se muestra en la figura 4.5 (a), o de 3 octetos, si se utiliza el tipo comprimido (*BNEP_COMPRESSED_ETHERNET*), tal y cómo se refleja en la figura 4.5 (b). Este último formato se emplea cuando el destinatario final de los datos Ethernet es un dispositivo BNEP, es decir, cuando la dirección destino del paquete BNEP coincide con la asociada al canal L2CAP por el que se está transmitiendo. Dicho de otro modo, se utiliza el formato comprimido cuando el origen y el destino de la trama Ethernet coinciden con un par maestro-esclavo de una *piconet* Bluetooth.



(a) Paquete de tipo general



(b) Paquete de tipo comprimido

Figura 4.5: Formatos de las cabeceras de los paquetes BNEP [Bluetooth SIG03b]

Puesto que la trama Ethernet encapsula un datagrama IP que, a su vez, transporta un datagrama UDP o TCP, el retardo de transmisión a calcular será el retardo de transmisión del nivel de transporte, que para el caso de estudio presentado ha sido UDP.

El retardo de transmisión de N octetos sobre UDP, $t_U(N)$, vendrá dado por la ecuación 4.9, expresada en función del retardo de transmisión a nivel IP, $t_I(N)$:

$$t_U(N) = t_I(N + C_U) \quad (4.9)$$

donde $C_U = 8$, es el tamaño, en octetos, de la cabecera UDP. Por su parte, $t_I(N)$ representa el retardo de IP para transmitir un datagrama de N octetos. La definición de $t_I(N)$, dada en la ecuación 4.10, tiene en cuenta la fragmentación que se puede producir a nivel IP cuando se supera la MTU de BNEP, M'_B , que coincide con la longitud del máximo *payload* de una trama Ethernet, es decir, 1500 octetos.

En concreto, $t_I(N)$ puede expresarse como sigue:

$$t_I(N) = n_{fnf_I} \cdot t_{ACK}(M'_B + C_B + C_L) + t_{TX}(L_{ff_I} + C_I + C_B + C_L) \quad (4.10)$$

siendo:

- C_I : el número de octetos de la cabecera IP (20 octetos, teniendo en cuenta los campos obligatorios de la cabecera de IP).

- C_B : el número de octetos correspondientes a la cabecera añadida por el protocolo BNEP, que viene dada por la ecuación siguiente.

$$C_B = \begin{cases} 15 & \text{si el tipo es general} \\ 3 & \text{si el tipo es comprimido} \end{cases} \quad (4.11)$$

- C_L : tal y como se definió en el apartado anterior, representa el tamaño de la cabecera de la PDU de L2CAP.
- $n_{fnf_I} = \left\lceil \frac{N}{M'_B - C_I} \right\rceil - 1$: el número de fragmentos intermedios en los que IP tiene que fragmentar el datagrama UDP para que, al añadirle la cabecera IP, no se exceda la MTU M'_B (véase la figura 4.6).

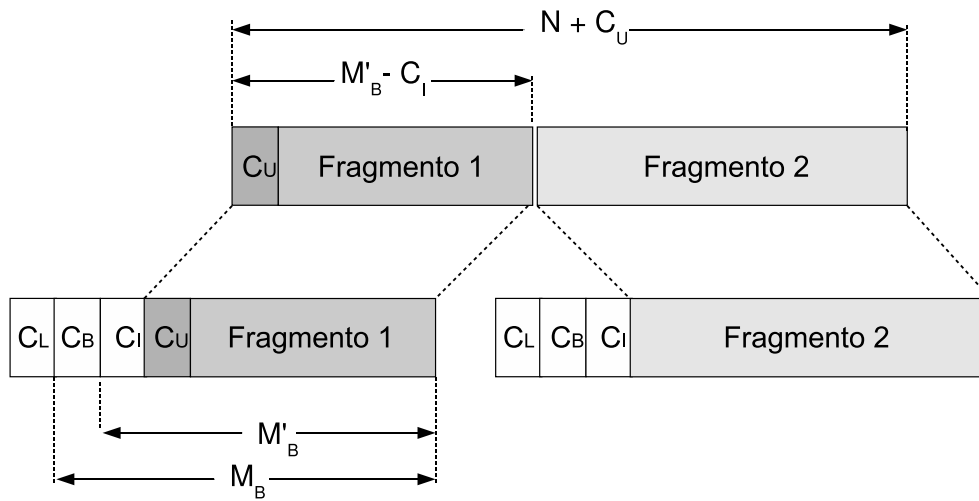


Figura 4.6: Fragmentación realizada en el nivel IP para que, al añadirle la cabecera IP, no se exceda la MTU M'_B

- $L_{ff_I} = ((N - 1) \bmod (M'_B - C_I)) + 1$: la longitud del fragmento final obtenido tras la fragmentación realizada a nivel IP para no exceder la MTU M'_B .

Obsérvese que hay que tener en cuenta que si el número de octetos del datagrama IP supera la MTU de BNEP, M'_B , se requerirá fragmentar el datagrama, de forma que por cada fragmento resultante se introducirán 20 octetos adicionales de cabecera IP.

De nuevo, nótese que si el datagrama IP cabe en un único fragmento BNEP (lo que sucede cuando el valor de N no excede $M'_B - C_I$), se tendrá que $n_{fnf_I} = 0$ y $L_{ff_I} = N$. Es decir, los datos se transmitirán como un único fragmento final.

Por último, cabe señalar que el protocolo BNEP no ha de realizar fragmentación de las tramas que entrega al nivel L2CAP para su transmisión, debido a que la especificación [Bluetooth SIG03b]:

1. Fija el valor mínimo de la MTU de L2CAP para BNEP, M_B , a 1691 octetos, según la expresión siguiente:

$$M_B = M'_B(1500 \text{ octetos}) + \text{cabecera BNEP}(15 \text{ octetos}) + \quad (4.12) \\ + \text{posible cabecera extendida} = 1691$$

2. Establece que el máximo *payload* que aceptará BNEP del nivel superior tiene que ser igual al valor de la MTU de L2CAP (valor mínimo: 1691), menos 191 octetos reservados para cabeceras BNEP.

El objetivo, tal y como se aclara en [Bluetooth SIG03b], es que los 1691 octetos, sumados a los 4 octetos de cabecera de L2CAP, puedan transportarse en cinco paquetes DH5 (de 339 octetos cada uno).

4.1.5. Comparación del retardo mínimo con el de los perfiles SPP y PAN

Es conveniente comenzar aclarando que la comparación de los retardos de los perfiles SPP y PAN se ha llevado a cabo teniendo en cuenta los valores de MTU utilizados en la implementación de los niveles del *Host* de la arquitectura Bluetooth empleada para la validación empírica (véase la sección 5.2). En esta implementación se fijan:

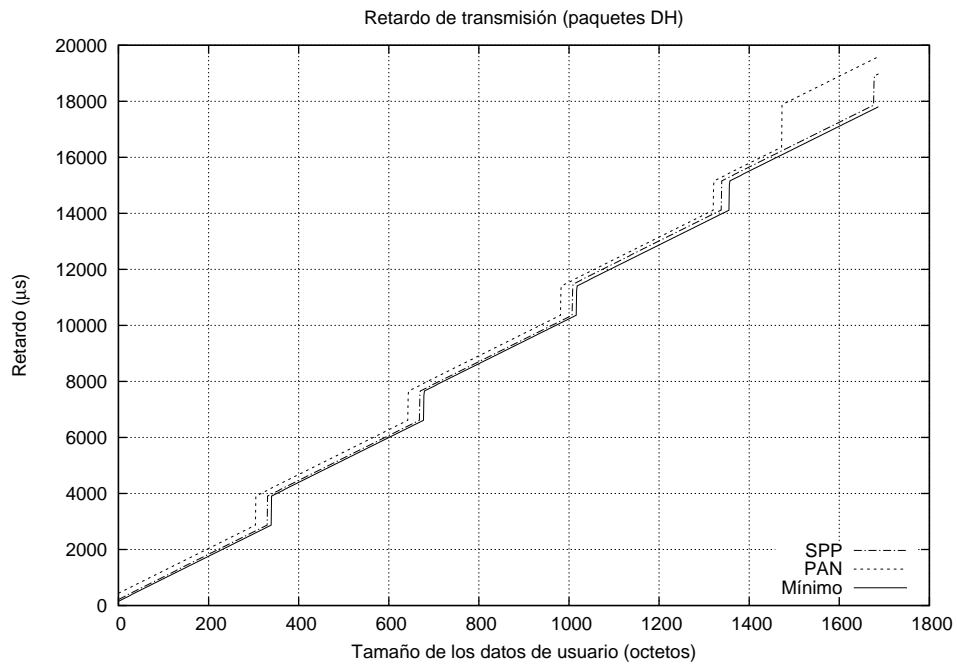
- La MTU de L2CAP para BNEP, M_B , al valor mínimo establecido por defecto en la especificación [Bluetooth SIG03b], esto es, 1691 octetos.
- La MTU de L2CAP para RFCOMM, M_R , a 1013 octetos.
- El parámetro $N1$ a 1008 octetos, decisión de implementación que permite evitar el efecto del *segmentation mismatch*.

La figura 4.7 muestra, para la transmisión de un dispositivo que actúa como maestro, la comparación entre el retardo teórico calculado teniendo en cuenta el *overhead* introducido para cada uno de los perfiles PAN y SPP, utilizando paquetes DH y DM, en el rango de longitudes de datos finales de usuario comprendidas entre 10 y 1500 octetos.

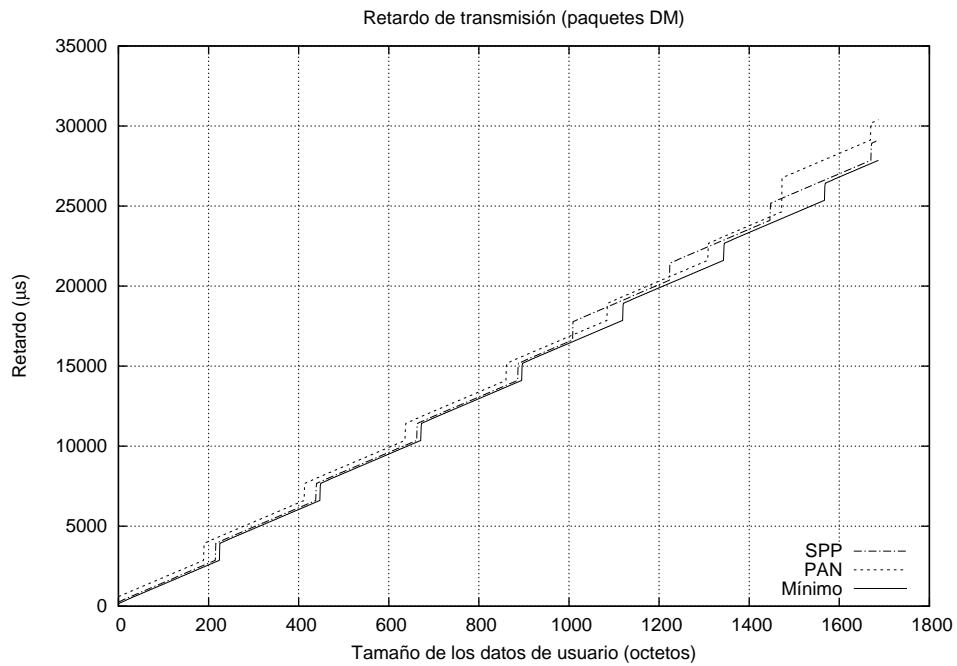
En la figura también se representa el retardo mínimo teórico de transmisión de datos de usuario, calculado a partir de la expresión 4.2, es decir, sin emplear perfiles, y evitando la inclusión de *overhead* de cada uno los protocolos de nivel superior.

Los escalones equiespaciados que aparecen en las figuras se corresponden con el llenado de paquetes de 5 *slots*, ya que una vez que la transmisión de una

4.1 Perfiles SPP y PAN: Efecto del *overhead* y del *segmentation mismatch* en el retardo



(a) Paquetes DH



(b) Paquetes DM

Figura 4.7: Comparación del retardo mínimo ($t_{TX}(N)$) con el correspondiente a los perfiles PAN ($t_U(N)$) y SPP ($t_R(N)$)

PDU de L2CAP requiere el uso de más de un paquete de 5 *slots*, tras el envío de cada fragmento en un paquete de 5 *slots* hay que esperar la confirmación del extremo remoto antes de iniciar la transmisión del siguiente fragmento. Obsérvese que para SPP estos saltos son prácticamente indistinguibles del caso ideal (sin *overhead*), porque el *overhead* introducido por SPP y L2CAP es muy reducido. Sin embargo, en el caso del perfil PAN los escalones tienen lugar antes. Por ejemplo, para paquetes DH, el incremento se produce para tamaños de $304 = 339 - 35$ octetos (y sus correspondientes múltiplos) debido al *overhead* añadido por UDP (8) + IP (20) + BNEP (3) + L2CAP (4) octetos.

Adicionalmente, puede observarse cómo el retardo de transmisión en el caso de SPP, cuando se emplean paquetes DM, supera al del perfil PAN a partir de la longitud de 1008 octetos que, con el *overhead* de RFCOMM, excede la MTU de L2CAP. A partir de este valor, el envío de más datos requiere la utilización de un nuevo fragmento RFCOMM, lo que se traduce en un incremento del retardo. Por contra, en caso de usar paquetes DH, el comportamiento es distinto porque la transmisión de 1008 octetos de datos de usuario, sumados a los 9 octetos de *overhead* puede realizarse en 3 paquetes DH5 ($1017/339$). Para ilustrar esta diferencia, es conveniente analizar la transmisión de 1010 octetos de usuario que, al rebasar el límite de los 1008 octetos, han de fragmentarse en 2 tramas de RFCOMM de 1008 y 2 octetos: la primera trama RFCOMM se transmite de forma óptima en tres paquetes DH5, y el residuo de 2 octetos, en un paquete DH1, lo que supone un tiempo total de 19 *slots*. Este tiempo es inferior a los 21 *slots* que tarda la transmisión utilizando BNEP, ya que los 1045 octetos, incluyendo el *overhead*, se envían en 3 paquetes DH5 y un paquete DH3. Sin embargo, en el caso de utilizar paquetes DM, el envío de la primera trama RFCOMM

precisa 4 paquetes DM5 y un paquete DM3, de forma que la segunda trama de RFCOMM, con los 2 octetos restantes, ha de esperar 28 *slots* para ser transmitida en un paquete DM1, pese que a esos 2 octetos podrían haberse transportado en el último paquete DM3 (si no hubiera sido necesario fragmentar a nivel RFCOMM). En cambio, con BNEP, debido a que no se realiza fragmentación, bastan 4 paquetes DM5 y un paquete DM3.

No obstante, es conveniente resaltar que, en el caso de RFCOMM, no se produce el *segmentation mismatch* [Okura02], que tendría lugar si no se hubiera adoptado la decisión de diseño de fijar M_R a un valor de, al menos, $N1 + 5$, o bien si algún dispositivo con el que se estableciera una comunicación RFCOMM soportara un valor de $N1$ inferior a 1008 octetos.

Finalmente, en el caso de BNEP, se puede observar que para datos de usuario de longitudes superiores a 1472 octetos aumenta el retardo debido a que, para este valor, es necesario que IP lleve a cabo fragmentación, al excederse los 1500 octetos que impone la MTU de BNEP, M_B (1472 octetos de datos + 28 de cabeceras IP y UDP).

4.1.6. Estimación del *throughput*

El *throughput* máximo teórico alcanzable, $Thr_{MAX}(N)$, en la transmisión de N octetos de usuario con un determinado paquete Bluetooth, asumiendo que no se realiza ninguna retransmisión y sin tener en cuenta el *overhead* añadido a los bits transmitidos, viene dado por la expresión siguiente:

$$Thr_{MAX}(N) = \frac{n_{b_u}(N)}{(n_{s_u} + 1) \cdot T_S} \quad (4.13)$$

donde:

- $n_{b_u}(N)$: es el número de bits de información útiles a transmitir para N octetos de usuario, es decir, excluyendo la información de redundancia, FEC y CRC, así como la cabecera del *payload*:

$$n_{b_u}(N) = N \cdot 8 \quad (4.14)$$

- n_{s_u} : es el número de *slots* ocupados por el paquete que se utiliza en la transmisión de los $n_{b_u}(N)$ de información, es decir 1, 3 ó 5 *slots*.

La tabla 4.1 indica el máximo *throughput* teórico alcanzable, calculado de acuerdo con la expresión 4.13, para el máximo número de bits que puede transportar cada tipo de paquete ACL.

Tipo de paquete ACL	Tamaño (octetos)	Tamaño máximo (bits)	Thr_{MAX} (kb/s)
DM1	0–17	136	108,8
DH1	0–27	216	172,8
DM3	0–121	968	387,2
DH3	0–183	1464	585,6
DM5	0–224	1792	477,8
DH5	0–339	2712	723,2
AUX1	0–29	232	185,6

Tabla 4.1: *Throughput* máximo para los distintos tipos de paquetes ACL

Obsérvese que la expresión 4.13, además de no computar el *overhead*, sólo considera que los N octetos a transmitir no exceden la máxima capacidad de

un paquete del nivel BB (339 ó 224 octetos). Ahora bien, si se quiere prescindir de estas restricciones, el *throughput* real, $Thr(N)$, ha de contemplar, no sólo el *overhead*, sino el hecho de que la transmisión de datos de usuario de longitudes elevadas requiere el envío de más de un paquete del nivel BB y que el esquema de transmisión utilizado por Bluetooth es TDD. De esta forma, $Thr(N)$ se podría calcular según la expresión indicada a continuación:

$$Thr(N) = \begin{cases} \frac{n_{b_u}(N)}{(n_{slots}(N) + 2) \cdot T_S} & (n_{slots}(N) \bmod 2) = 0 \\ \frac{n_{b_u}(N)}{(n_{slots}(N) + 1) \cdot T_S} & (n_{slots}(N) \bmod 2) \neq 0 \end{cases} \quad (4.15)$$

donde $n_{slots}(N)$ es el número de *slots* requeridos para la transmisión de los $n_{b_u}(N)$ bits más su correspondiente *overhead*:

$$n_{slots}(N) = \left\lceil \frac{t_{TX_{TOTAL}}(N)}{T_S} + 0,4144 \right\rceil \quad (4.16)$$

En la expresión 4.16 el tiempo de transmisión $t_{TX_{TOTAL}}(N)$ corresponderá a $t_{TX}(N)$ (4.2), $t_R(N)$ (4.6) o $t_U(N)$ (4.9), dependiendo de que se esté considerando la transmisión directamente sobre el nivel BB o de que se tenga en cuenta el *overhead* añadido a los bits de información por cada uno de los niveles al utilizar los perfiles SPP o PAN.

Este tiempo de transmisión, $t_{TX_{TOTAL}}(t_{TX}(N), t_U(N)$ o $t_R(N)$), según el caso), se divide entre T_S , y se le añade una corrección para contemplar el hecho de que los paquetes no llenan el último *slot*, es decir, que existe un tiempo de guarda de $625\mu s - 366\mu s = 259\mu s$, que corresponde al 41,44 % ($259\mu s / 625\mu s$) de la duración

de un *slot* (véase el apartado 2.1.2.4). Por último, se redondea por exceso para obtener un número entero de *slots*.

Hay que resaltar que la ecuación 4.15 diferencia que el número de *slots* obtenidos mediante la expresión 4.16 sea par o impar, ya que en el primer caso hay que tener en cuenta que las transmisiones son de 1, 3 ó 5 *slots*. Esto es, dos *slots* realmente corresponden a tres, y cuatro *slots* han de computarse como cinco. Finalmente, al tiempo de transmisión total, múltiplo de T_S , se le suma el *slot* en el que el otro extremo confirma la recepción.

Las gráficas representadas en la figura 4.8 muestran el *throughput* teórico obtenido, de acuerdo con la ecuación 4.15, a partir del retardo teórico calculado para los perfiles SPP y PAN en los apartados 4.1.3 y 4.1.4, incluyendo el *overhead*, tanto para paquetes DH como para paquetes DM, en el rango de longitudes de datos finales de usuario comprendidas entre 10 y 1500 octetos. Puede apreciarse que, cuando se emplean paquetes DM, el *throughput* de SPP es mejor que el de PAN para longitudes inferiores a 1008 octetos, punto a partir del cual, al añadir el *overhead* de RFCOMM, se supera la MTU de L2CAP. Sin embargo, el comportamiento con paquetes DH es distinto ya que, como se explicó en el apartado anterior, la transmisión de 1008 octetos se realiza, de forma óptima, con 3 paquetes DH5.

En la figura 4.8 también se muestra:

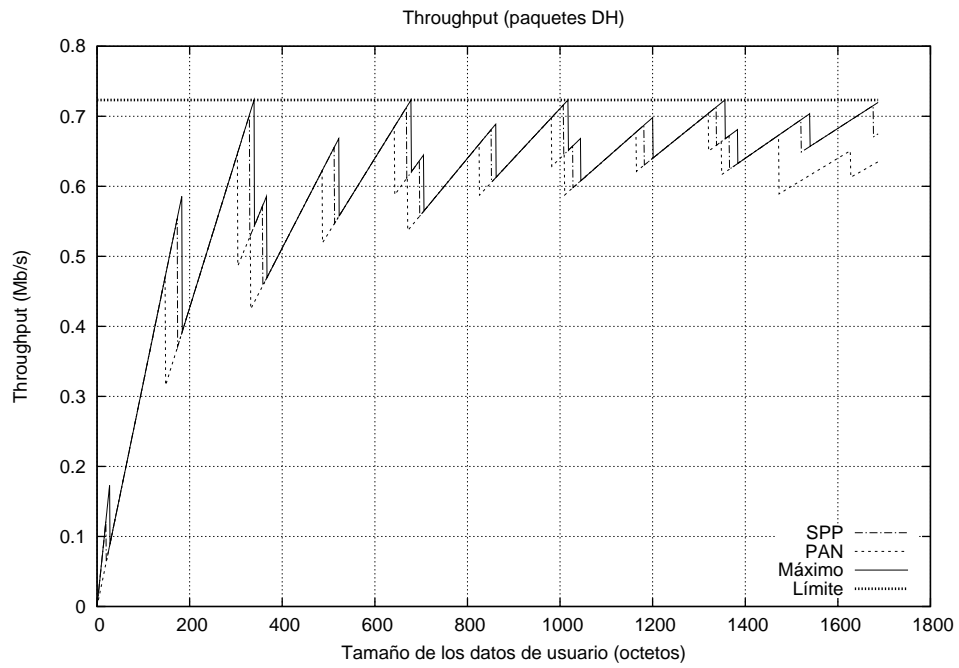
- El *throughput* máximo teórico, calculado identificando en la ecuación 4.16 el tiempo de transmisión $t_{TX_{TOTAL}}$ con el tiempo definido por la expresión 4.2, en vez de hacerlo con t_U o t_R , como se ha hecho para los perfiles PAN o SPP. Así, la gráfica etiquetada como «Máximo» representa el *throughput*

obtenido sin emplear perfiles y, evitando la fragmentación y la inclusión de *overhead* de cada uno los protocolos de nivel superior.

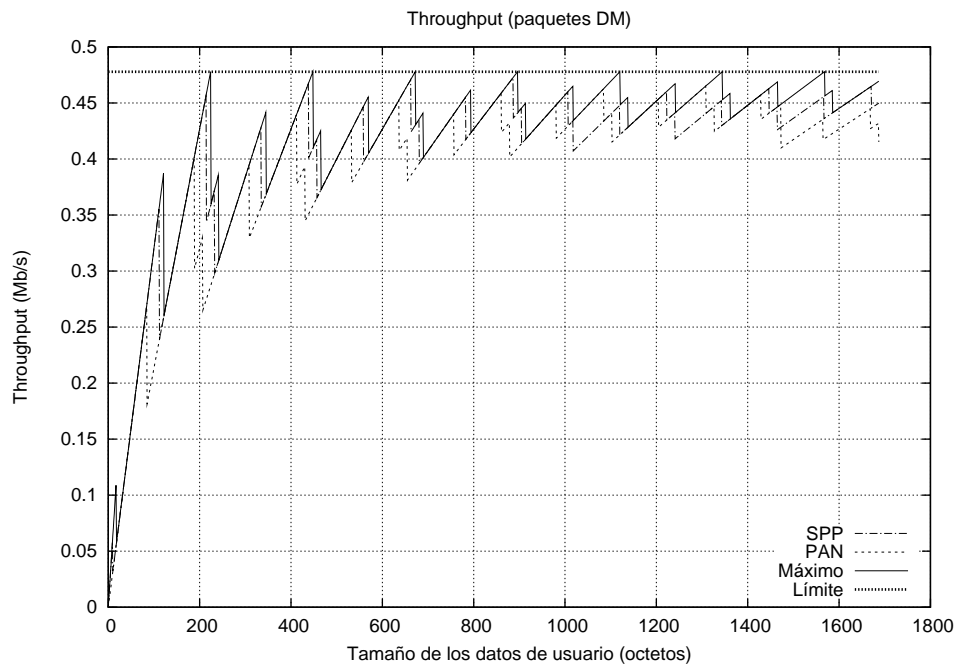
- La cota máxima de Thr_{MAX} (o «Límite»), alcanzable con los paquetes DH5 y DM5, que como se indica en la tabla 4.1 corresponde, respectivamente, a una tasa de 723,2 kb/s y 477,8 kb/s.

Finalmente, con el objeto de destacar la importancia del *segmentation mismatch* en el perfil SPP, se muestra cómo disminuye el *throughput* en dos casos:

- Cuando el parámetro $N1$ no se establece a un valor superior a la máxima longitud de datos a transmitir. Con este fin, en la figura 4.9 (a) se representa el *throughput* para distintos valores de $N1$, y fijando M_R de forma que se satisfice la inecuación $M_R \geq N1 + 5$ para el máximo valor de $N1$, 1500 octetos, caso que corresponde al máximo rendimiento porque se cumple que $N1 \geq N$.
- Cuando no se satisfice la relación $M_R \geq N1 + 5$. En esta ocasión, en la figura 4.9 (b) se ha representado el *throughput* para distintos valores de M_R , de los cuales uno de ellos es el valor definido por defecto para la MTU de L2CAP (672 octetos). El valor de $N1$ se ha fijado a $672 - 5 = 667$ octetos. Puede observarse que el caso óptimo, tiene lugar cuando se cumple que $M_R = N1 + 5$ y que se produce una pérdida de rendimiento apreciable, conforme M_R decrece en relación con $N1$. En particular, nótese que disminuir el valor de M_R de forma que no satisfaga la condición anterior, $M_R \geq N1 + 5$, ocasiona caídas del *throughput* de más de un 25 %.



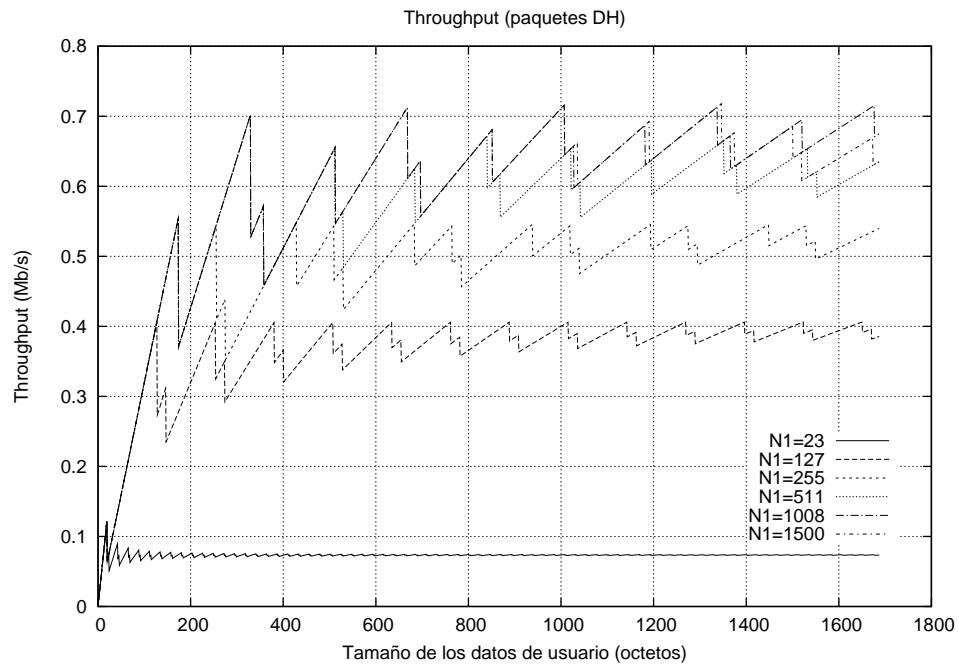
(a) paquetes DH



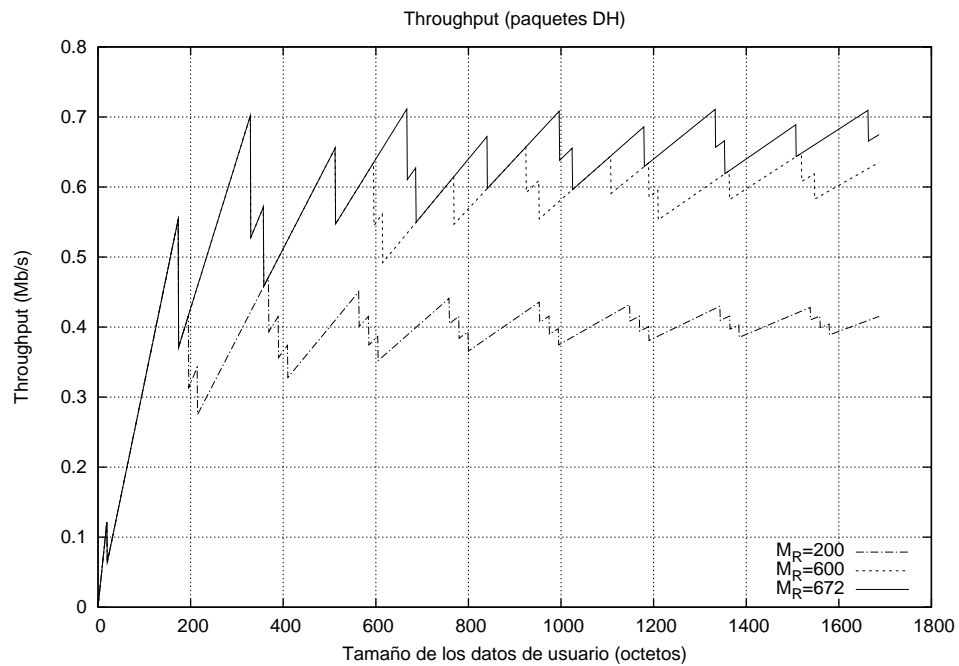
(b) Paquetes DM

Figura 4.8: Comparación del *throughput* máximo con el correspondiente a los perfiles PAN y SPP

4.1 Perfiles SPP y PAN: Efecto del *overhead* y del *segmentation mismatch* en el retardo



(a) Fijando M_R a 1505 octetos



(b) Fijando N_1 a 667 octetos

Figura 4.9: Efecto del *segmentation mismatch* sobre el *throughput*, para paquetes DH cuando se emplea el perfil SPP

4.1.7. Discusión

Aunque el empleo del perfil PAN o SPP vendrá dado por las características de la aplicación, es recomendable tener presente el hecho de que la transmisión de datos de usuario mediante el perfil PAN, en principio, implica un menor rendimiento a consecuencia del *overhead* introducido por TCP/IP (o UDP/IP). Otro factor crítico a considerar es el máximo tamaño manejable por los protocolos BNEP y RFCOMM, que es de 1500 octetos en BNEP (máximo tamaño de una trama *Ethernet*) y hasta de 32767 en RFCOMM. No obstante, este dato aislado no es significativo si se obvia la relación con la MTU de L2CAP. De hecho, cabe resaltar la pérdida de rendimiento que se produce cuando no se tiene en cuenta las máximas MTUs de cada uno de los niveles y las relaciones existentes entre las de capas adyacentes.

En el caso de PAN, la especificación de BNEP [Bluetooth SIG03b] establece el valor de la MTU de L2CAP y, en función de ésta y del máximo tamaño de la cabecera de BNEP, limita la longitud de la trama que el nivel superior podrá entregar a BNEP, garantizando así que no se produce el efecto del *segmentation mismatch*, véase expresión (4.12). Sin embargo, no ocurre lo mismo en el caso de la especificación de SPP.

Del estudio realizado, se concluye que para minimizar el efecto del *segmentation mismatch* en SPP, el parámetro $N1$ negociado entre entidades RFCOMM homólogas tiene que satisfacer las siguientes relaciones²:

- Respecto al tamaño de los datos (N) que se van a transmitir: $N1 \geq N$.

²Recuérdese que, tal y como se explicó en el apartado 4.1.4, el protocolo BNEP no tiene que fragmentar las tramas que entrega al nivel L2CAP, por lo que no se precisa tener en cuenta restricción adicional alguna.

- Respecto a la MTU de L2CAP: $M_R \geq N1 + 5$.

De no cumplirse estas inecuaciones, el rendimiento puede caer por debajo del que se consigue utilizando el perfil PAN, basado en BNEP, para el que, en primera instancia, se logra un peor rendimiento derivado del excesivo *overhead* introducido por la cabeceras de UDP/IP y BNEP.

En cualquier caso, una ventaja de RFCOMM frente a BNEP a tener presente es que el límite superior del parámetro $N1$ es 32767 octetos y que se puede cumplir sin problemas la condición anterior, ya que la máxima longitud del *payload* de una PDU L2CAP es de 65535 octetos. Así, fijando ese máximo, la fragmentación sólo se produciría a nivel BB. Sin embargo, en el caso de BNEP, para tamaños de datos de usuario superiores a 1472 (1500 – 20 – 8) octetos, empeora el rendimiento ya que se requeriría fragmentación a nivel IP (puesto que BNEP impone una MTU de 1500 octetos), añadiéndose, por cada fragmento, 20 octetos de *overhead*, además de la correspondiente cabecera de BNEP, que puede ser de hasta 15 octetos, y de la cabecera de L2CAP.

Finalmente, en la figura 4.10 se representa, para longitudes superiores a 1000 octetos, el retardo teórico mínimo en el caso de SPP, cuando la MTU de L2CAP para RFCOMM (M_R) y el parámetro $N1$ se establecen a un valor superior a M_B , acorde con la máxima longitud de datos a transmitir, $N1 \geq N$, y satisfaciendo la inecuación $M_R \geq N1 + 5$. Nótese que de esta forma el rendimiento de RFCOMM es mejor que el que se logra con BNEP para todos los tamaños, especialmente para aquellos superiores a 1472 octetos.

No obstante, es conveniente aclarar que con la implementación empleada para la realización de las medidas descritas en 5.2, no es posible configurar el

valor de la MTU de L2CAP para RFCOMM ni el parámetro $N1$ por encima de 1013 octetos y de 1008 octetos, respectivamente. Por consiguiente, el perfil PAN, a pesar de introducir más cabeceras, en la práctica puede alcanzar un mayor *throughput*.

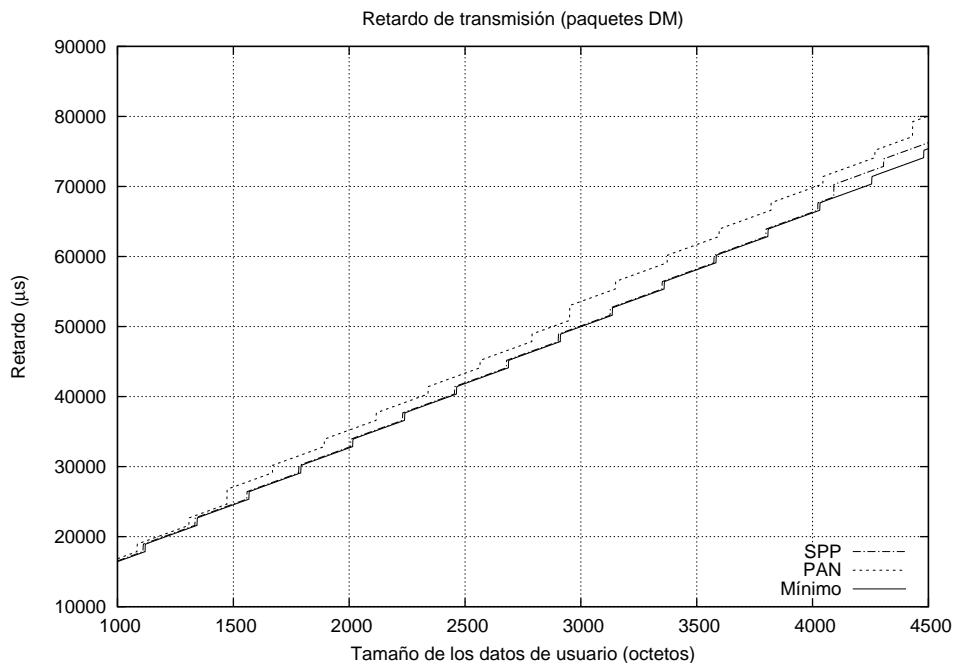


Figura 4.10: Comparación del retardo para los perfiles PAN y SPP, con paquetes DM, fijando M_R y $N1$, a 4096 y 4091 octetos, respectivamente, satisfaciéndose así las condiciones $N1 \geq N$ y $M_R \geq N1 + 5$

4.2. Perfil SPP: Efecto de la configuración de calidad de servicio en el retardo de transmisión

El objetivo de esta sección es analizar el efecto que el procedimiento de *poll* causa sobre el retardo de transmisión. Este proceso fija un intervalo (*poll interval*) que, como ya se ha explicado en el apartado 2.1.3, es el máximo tiempo entre transmisiones sucesivas del maestro a un determinado esclavo en enlaces ACL, y permite reservar ancho de banda así como controlar la latencia de la transmisión. El intervalo de *poll* se garantiza en el modo activo (*ACTIVE*), excepto cuando se producen colisiones con la realización de los procedimientos asociados a los subestados *page*, *page scan*, *inquiry* e *inquiry scan*.

Aunque en las especificaciones de Bluetooth (parte C de la especificación [Bluetooth SIG01c]) el intervalo de *poll*, referido también como T_{poll} , se define utilizando como unidad de medida el *slot*, en lo sucesivo, se utilizará n_{poll} y T_{poll} para hacer referencia al intervalo de *poll* expresado, respectivamente, en número de *slots* y en microsegundos, de forma que:

$$T_{poll} = n_{poll} \cdot T_S \quad (4.17)$$

Como ya se anticipó en el apartado 2.1.3, en las especificaciones de Bluetooth (parte C de la especificación [Bluetooth SIG01c]) se indica que una unidad que esté actuando como maestro sólo está obligada a comunicarse con cada esclavo de la *piconet* una vez cada n_{poll} *slots*.

Para incorporar en el modelo desarrollado el esquema de comunicación regido por el proceso de *poll*, en una primera aproximación, podría pensarse que una corrección consistiría en sumar al valor obtenido por la ecuación 4.6, el tiempo correspondiente a los n_{poll} *slots* que, en el peor de los casos, tarda el maestro en direccionar al esclavo. Con esta modificación el retardo para el perfil SPP, en el sentido de transmisión maestro-esclavo, vendría dado por la siguiente ecuación:

$$t_{R_M}(N) = T_{poll} + t_R(N) \quad (4.18)$$

Sin embargo, la ecuación anterior no contempla el impacto del intervalo de *poll* sobre el retardo de transmisión cuando los datos de usuario y el *overhead* añadido por cada una de los niveles excede la máxima capacidad de un paquete DH5. Esto es, hay que tener en cuenta que los fragmentos intermedios de una PDU de L2CAP podrían estar separados hasta un máximo de n_{poll} *slots*. La razón de este comportamiento tiene un matiz diferente en cada sentido de comunicación:

- **Sentido de transmisión maestro-esclavo:** La confirmación del fragmento intermedio k -ésimo transmitida por el esclavo se ha de efectuar en el siguiente *slot*, pero la transmisión del fragmento $(k + 1)$ -ésimo, en el peor de los casos, podría demorarse hasta n_{poll} *slots*, contados desde el inicio de la transmisión del maestro, tal y como se muestra la figura 4.11.
- **Sentido de transmisión esclavo-maestro:** Cuando el esclavo envía un paquete que contiene el fragmento intermedio k -ésimo, ha de esperar la

4.2 Perfil SPP: Efecto de la configuración de calidad de servicio en el retardo de transmisión

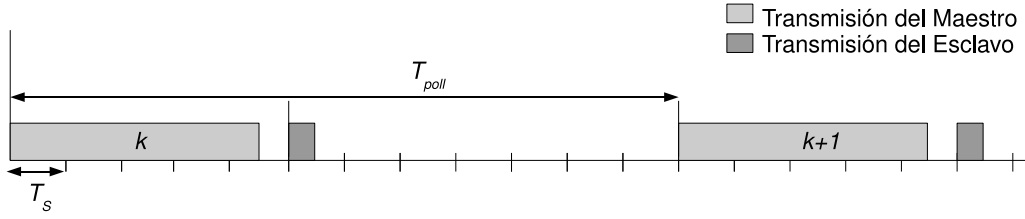


Figura 4.11: Efecto del intervalo de *poll* en la transmisión de fragmentos consecutivos para el sentido de comunicación maestro-esclavo

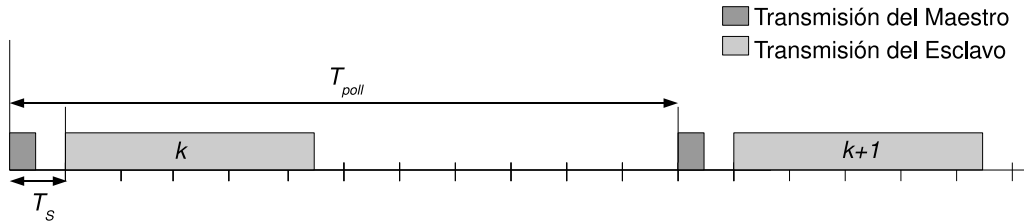


Figura 4.12: Efecto del intervalo de *poll* en la transmisión de fragmentos consecutivos para el sentido de comunicación esclavo-maestro

confirmación del extremo remoto, hasta un tiempo máximo de T_{poll} , antes de iniciar la transmisión del siguiente fragmento (véase la figura 4.12).

Consecuentemente, en el caso peor, para ambos sentidos de transmisión la componente del retardo $t_{ACK}(N)$, notada por $t_{ACK_{poll}}(N)$, tiene que ser redefinida como se indica en la siguiente ecuación:

$$t_{ACK_{poll}}(N) = \begin{cases} 0 & N = 0 \\ T_{poll} & 0 < N \leq L_5 \\ T_{poll} \cdot \left\lfloor \frac{N}{L_5} \right\rfloor + \\ + t_{ACK_{poll}}(N \bmod L_5) & N > L_5 \end{cases} \quad (4.19)$$

Asimismo, la componente $t_{TX}(N)$, notada por $t_{TX_{poll}}(N)$, vendrá dada por

la siguiente expresión:

$$t_{TX_{poll}}(N) = \begin{cases} 0 & N = 0 \\ n_b(N) \cdot t_b & 0 < N \leq L_5 \\ t_{ACK_{poll}}(L_5) \cdot \left\lfloor \frac{N}{L_5} \right\rfloor + & N > L_5 \\ +t_{TX_{poll}}(N \bmod L_5) & \end{cases} \quad (4.20)$$

La inclusión de las componentes 4.19 y 4.20 en la ecuación de retardo 4.6 permite obtener una cota máxima del retardo para el perfil SPP en el sentido de transmisión maestro-esclavo, $t_{R_M}(N)$, definida según la ecuación:

$$t_{R_M}(N) = T_{poll} + n_{fnf_R} \cdot t_{ACK_{poll}}(L_R + O_R(L_R) + C_L) + \quad (4.21) \\ +t_{TX_{poll}}(L_{ff_R} + O_R(L_{ff_R}) + C_L)$$

En el sentido de comunicación esclavo-maestro hay que tener en cuenta que el dispositivo que actúa como esclavo sólo puede iniciar la transmisión inmediatamente después de haber sido direccionado por el maestro en el *slot* anterior. En consecuencia, la cota del retardo de transmisión, $t_{R_E}(N)$, puede obtenerse a partir de la definida para el maestro, sumando el *slot* de direccionamiento, como se indica a continuación:

$$t_{R_E}(N) = T_S + t_{R_M}(N) \quad (4.22)$$

Nótese que el *slot* en el que es direccionado el esclavo se computa en la ecuación 4.22 y no en la definición de 4.20. Esto es debido a que una vez que el

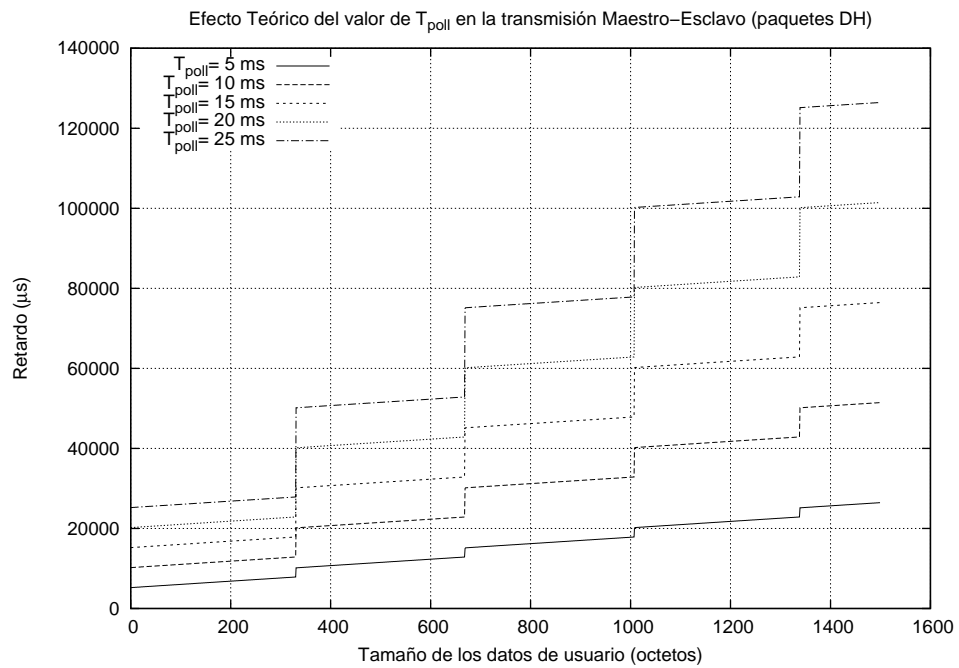
esclavo es direccionado y transmite el primer paquete de datos, las siguientes transmisiones se van a realizar, a lo sumo, cada n_{poll} slots.

Es conveniente aclarar que las ecuaciones 4.21 y 4.22 representan cotas superiores porque en caso de que el maestro sólo tuviera establecida una conexión con un esclavo y no tuviera otras tareas que atender, el maestro podría continuar direccionando al esclavo, sin esperar n_{poll} slots:

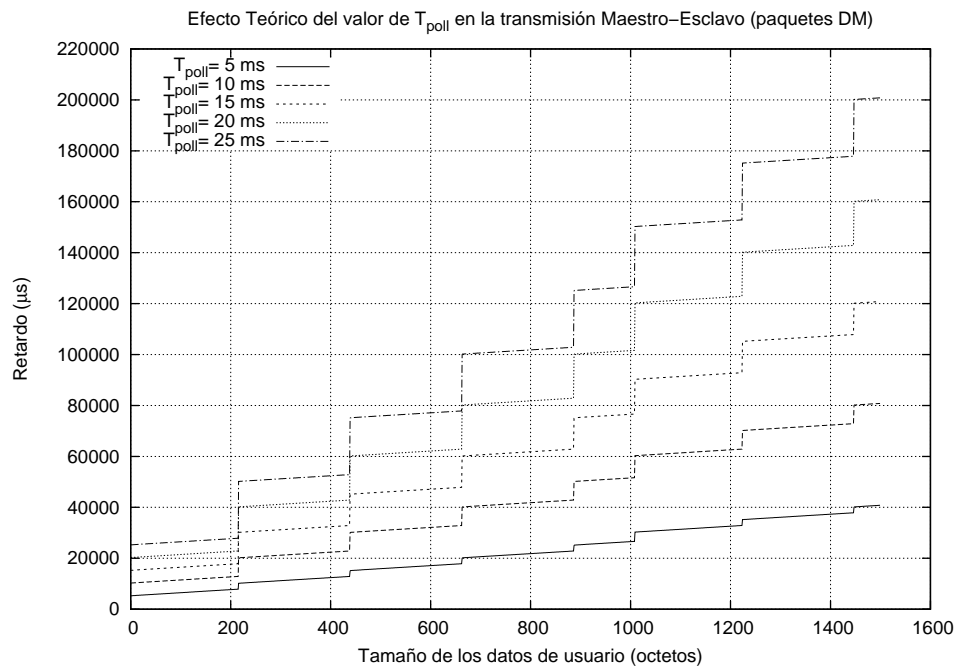
- **Sentido maestro-esclavo:** mientras el maestro tuviera datos pendientes de envío.
- **Sentido esclavo-maestro:** siempre que el esclavo enviase al maestro paquetes de datos, o lo que es lo mismo, hasta que el esclavo, al no tener más datos pendientes de transmisión, enviara un paquete NULL.

Las figuras 4.13 y 4.14 representan los retardos obtenidos para el perfil SPP, calculados, respectivamente, por las ecuaciones 4.21 y 4.22 para distintos valores de T_{poll} y diferenciando entre paquetes DH y DM, en los sentidos de transmisión maestro-esclavo y esclavo-maestro. Obsérvese que, cuanto más alto es el valor de T_{poll} , y más paquetes de 5 slots se requieren para la transmisión de una PDU de L2CAP, mayores son los sucesivos escalones equiespaciados que presenta la evolución del retardo.

A partir de estas figuras se puede deducir que la cota del retardo definida por las ecuaciones 4.21 y 4.22, si se configura T_{poll} a un valor elevado, alcanza valores que pueden degradar notablemente las prestaciones de flujos de tiempo real.



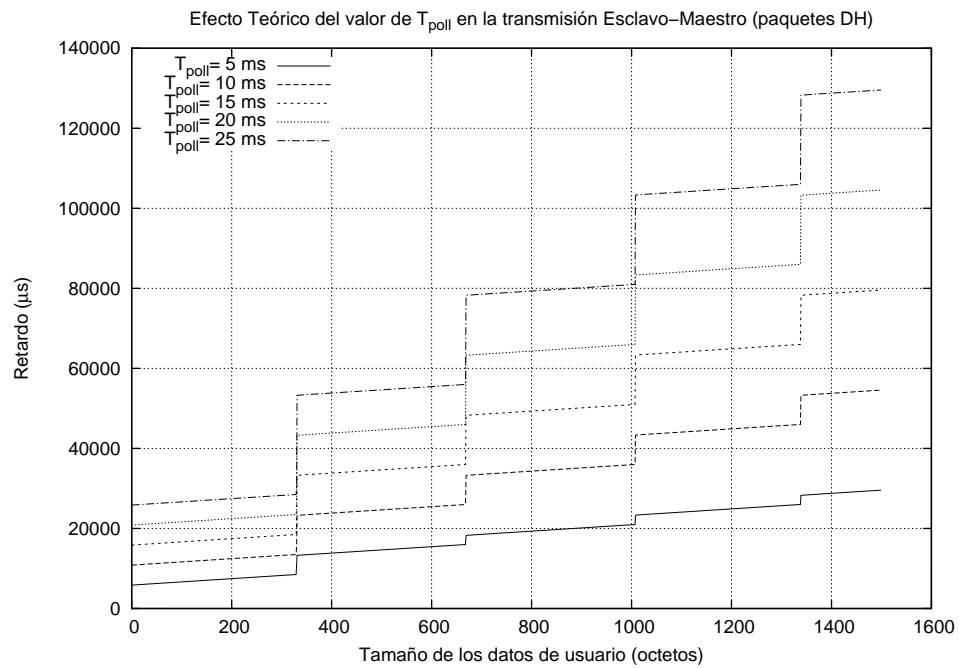
(a) Paquetes DH



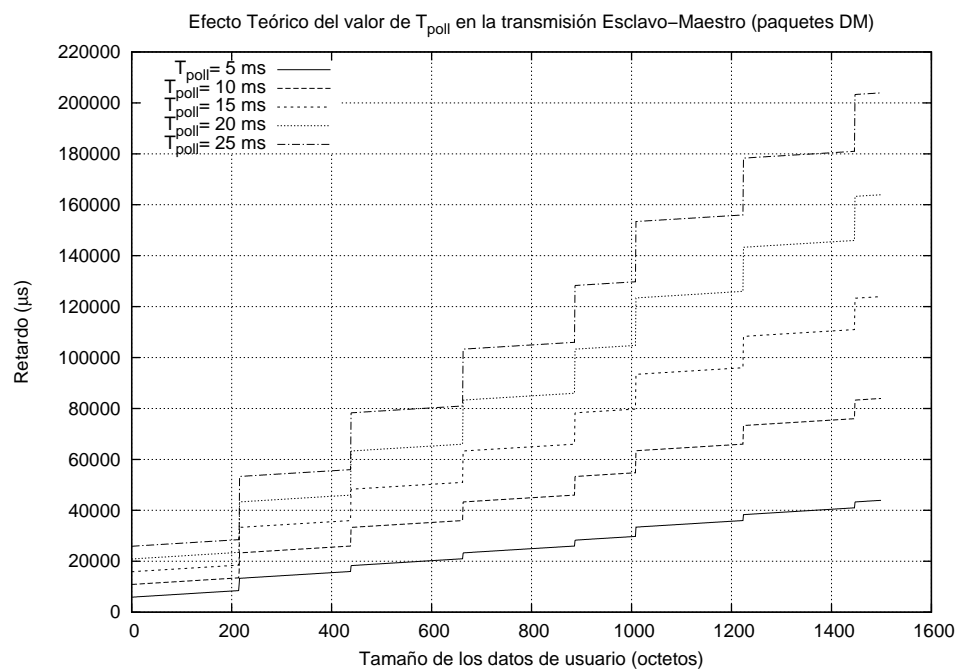
(b) Paquetes DM

Figura 4.13: Retardo máximo de transmisión en el sentido maestro-esclavo para el perfil SPP, teniendo en cuenta el efecto del intervalo de *poll*

4.2 Perfil SPP: Efecto de la configuración de calidad de servicio en el retardo de transmisión



(a) Paquetes DH



(b) Paquetes DM

Figura 4.14: Retardo máximo de transmisión en el sentido esclavo-maestro para el perfil SPP, teniendo en cuenta el efecto del intervalo de *poll*

4.3. Perfil SPP: Efecto de las retransmisiones en el retardo de transmisión

En esta sección se pretende extender el modelo anteriormente desarrollado a fin de incluir el efecto de las retransmisiones que puedan producirse a nivel BB para contrarrestar la aparición de errores en la transmisión Bluetooth. Al respecto, conviene aclarar que en el modo de funcionamiento del protocolo L2CAP de Bluetooth 1.1, que es equivalente al modo de operación básico del protocolo L2CAP de Bluetooth 1.2 (modo por defecto), no se realizan retransmisiones a nivel L2CAP.

El hecho de contemplar el efecto de las retransmisiones a nivel BB en el cálculo del retardo implica la modificación de las ecuaciones correspondientes a las dos componentes, $t_{ACK}(N)$ y $t_{TX}(N)$, en función de las cuales se definió el retardo mínimo en la sección 4.1. Los valores medios esperados de estas componentes se definirán en función de la probabilidad de retransmisión de un paquete, entendida como la probabilidad de que el paquete tenga errores no recuperables, y se denotarán respectivamente como $\overline{t_{ACK_{per}}}(N)$ y $\overline{t_{TX_{per}}}(N)$. Con este propósito, se comienza definiendo la tasa de paquetes erróneos, PER, expresada en función de la tasa de bits erróneos, BER.

La tasa PER, para paquetes DH, viene dada por la probabilidad $P(E)$ de que exista, al menos, un bit erróneo:

$$P(E) = P(X \geq 1) = 1 - P(X = 0) \quad (4.23)$$

siendo la variable X el número de bits erróneos en un paquete.

Asumiendo, por simplificar, que los errores se producen de forma incorrelada entre sí, la probabilidad de que haya i errores en la transmisión de un paquete se puede caracterizar mediante una distribución binomial dada por:

$$P(X = i) = \binom{n}{i} \cdot p^i \cdot (1 - p)^{n-i} \quad (4.24)$$

donde n representa el número de bits de los que consta el paquete y p es la probabilidad de que un bit sea erróneo.

Particularizando la ecuación 4.23 para la variable aleatoria binomial indicada con parámetros $p = BER$ y $n = n_d(N)$ (véase la ecuación 4.4), se obtiene la tasa PER para paquetes DH, expresada en función de la tasa BER:

$$\begin{aligned} PER_{DH_d}(N) &= P(X \geq 1) = 1 - P(X = 0) \\ &= 1 - (1 - BER)^{n_d(N)} \end{aligned} \quad (4.25)$$

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 modificar la ecuación 4.25 teniendo en cuenta que se trata de un código de Hamming (15,10) recortado (*shortened*).

Dado un código de Hamming (n, k), la probabilidad de que la decodificación de un bloque de n bits sea errónea está acotada por la siguiente ecuación [Moon05]:

$$P(E) \leq \sum_{i=t+1}^n \binom{n}{i} \cdot BER^i \cdot (1 - BER)^{n-i} \quad (4.26)$$

donde:

- $n = 2^m - 1$: es el número de símbolos de salida, es decir, la longitud de la palabra codificada.
- $k = 2^m - m - 1$: es el número de símbolos de entrada, o símbolos de información.
- $m = n - k$: es el número de símbolos de paridad.
- $t = \left\lfloor \frac{d^* - 1}{2} \right\rfloor$: es la capacidad de corrección de errores aleatorios.
- d^* : es la distancia de Hamming más pequeña entre las distintas palabras de código, siendo la distancia de Hamming entre dos secuencias, el número de bits en que se diferencian.

En el caso particular del código (15,10), por cada 10 bits se añaden 5 bits de redundancia, de forma que un grupo de 15 bits es correcto si hay un bit erróneo como máximo ($t = 1$). Por consiguiente, la probabilidad de que un bloque se pierda, P_b , vendrá dada por:

$$\begin{aligned}
 P_b &\leq \sum_{i=2}^{15} \binom{15}{i} \cdot BER^i \cdot (1 - BER)^{15-i} = \\
 &= 1 - \sum_{i=0}^1 \binom{15}{i} \cdot BER^i \cdot (1 - BER)^{15-i}
 \end{aligned} \tag{4.27}$$

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 que se pierda un paquete DM, $PER_{DM_d}(N)$, expresada en función de la probabilidad P_b , definida por la ecuación 4.27, se puede calcular a partir de la probabilidad de que no se pierda ninguno de los grupos de 15 bits que lo componen:

$$\begin{aligned}
 PER_{DM_d}(N) &= P(X \geq 1) = 1 - P(X = 0) = \\
 &= 1 - \binom{n_g(N)}{0} \cdot P_b^0 \cdot (1 - P_b)^{n_g(N)-0} \\
 &= 1 - (1 - P_b)^{n_g(N)}
 \end{aligned} \tag{4.28}$$

donde $n_g(N) = n_d(N)/15$, representa el número de grupos de 15 bits de los que consta un paquete DM, y $n_d(N)$ se obtiene a partir de la segunda rama de la ecuación 4.4, que tiene en cuenta que, además del campo de datos, se protege la cabecera del *payload* y los dos octetos de CRC.

Nótese que, en este caso, la variable aleatoria X , incluida en la ecuación 4.28, representa el número de veces que se recibe un bloque con errores no recuperables en los $n_g(N)$ bloques que constituyen el paquete DM.

Finalmente, sustituyendo 4.27 en la ecuación 4.28, resulta:

$$PER_{DM_d}(N) = 1 - \left((1 - BER)^{15} + 15 \cdot BER \cdot (1 - BER)^{14} \right)^{n_g(N)} \tag{4.29}$$

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].

Si, además, se tiene en cuenta que la cabecera de los paquetes Bluetooth también se protege con codificación FEC 1/3, siguiendo el mismo procedimiento

por el que se ha obtenido la ecuación 4.29, se concluye que la probabilidad de que se produzca un error irrecuperable en la cabecera, PER_c , viene dada por la ecuación:

$$PER_c = 1 - \left((1 - BER)^3 + 3 \cdot BER \cdot (1 - BER)^2 \right)^{18} \quad (4.30)$$

Nótese que, en la expresión anterior, se contempla que cada uno de los 18 bits de los que consta la cabecera sin codificar se transforman en 3, resultando un total de 54 bits, o lo que es lo mismo, 18 grupos de 3 bits. Obsérvese que, también en este caso, se ha tenido en cuenta que la protección permite corregir hasta 1 bit en cada bloque de 3 bits.

Así, la probabilidad de que se pierda un paquete, $PER(N)$, vendrá dada por la siguiente ecuación:

$$PER(N) = \begin{cases} PER_{DH_d}(N) + PER_c - PER_{DH_d}(N) \cdot PER_c & \text{paquetes DH} \\ PER_{DM_d}(N) + PER_c - PER_{DM_d}(N) \cdot PER_c & \text{paquetes DM} \end{cases} \quad (4.31)$$

En esta ecuación se refleja que la probabilidad de que se pierda un paquete, p , será la probabilidad de que se produzca un error irrecuperable en la cabecera o en el campo de datos, teniendo en cuenta que la probabilidad de la unión de dos sucesos se define como:

$$p(A \cup B) = p(A) + p(B) - p(A \cap B) \quad (4.32)$$

y que, cuando los sucesos A y B son independientes, se cumple que:

$$p(A \cap B) = p(A) \cdot p(B) \quad (4.33)$$

Una vez determinada la tasa de paquetes erróneos, PER , y, asumiendo que el valor del temporizador $flushTO$ se configura para realizar infinitas retransmisiones, el número medio de intentos necesarios para que un paquete sea transmitido con éxito, denotado por $\overline{N_{RTx}}$, se determina de acuerdo con la ecuación³:

$$\begin{aligned}
 \overline{N_{RTx}} &= 1 \cdot (1 - p) + 2 \cdot p \cdot (1 - p) + 3 \cdot p \cdot p \cdot (1 - p) + \dots = \\
 &= (1 - p) \cdot \sum_{i=1}^{\infty} i \cdot p^{i-1} = \\
 &= \frac{1}{1 - p}
 \end{aligned} \tag{4.34}$$

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

$$\begin{aligned}
 \overline{t_{ACK_{per}}}(N) &= (1 - p) \cdot t_{ACK}(N) + \\
 &+ p \cdot (1 - p) \cdot (t_{RTx}(N) + t_{ACK}(N)) + \dots \\
 &\dots + p^i \cdot (1 - p) \cdot (i \cdot t_{RTx}(N) + t_{ACK}(N))
 \end{aligned} \tag{4.35}$$

donde i representa el número de transmisiones fallidas que se producen antes de la que se lleva a cabo satisfactoriamente.

En la fórmula anterior, $t_{RTx}(N)$, que se definirá posteriormente, es el tiempo transcurrido desde que se inicia la transmisión de un paquete DM o DH hasta que se realiza la retransmisión.

³Para facilitar la legibilidad, en lo sucesivo, se utilizará el símbolo p para denotar la tasa de paquetes erróneos $PER(N)$ definida por la ecuación 4.31.

Reordenando los términos de la ecuación 4.35, resulta:

$$\begin{aligned}
 \overline{t_{ACK_{per}}}(N) &= (1-p) \cdot t_{ACK}(N) \cdot (1+p+p^2+p^3+\dots+p^i)+ \\
 &\quad + p \cdot (1-p) \cdot t_{RTx}(N) \cdot (1+2p+3p^2+4p^3+\dots+i \cdot p^i) = \\
 &= (1-p) \cdot t_{ACK}(N) \cdot \sum_{i=1}^{\infty} p^i + \\
 &\quad + p \cdot (1-p) \cdot t_{RTx}(N) \cdot \sum_{i=1}^{\infty} i \cdot p^{i-1}
 \end{aligned} \tag{4.36}$$

Finalmente, sumando las series, se obtiene:

$$\begin{aligned}
 \overline{t_{ACK_{per}}}(N) &= \frac{p}{1-p} \cdot t_{RTx}(N) + t_{ACK}(N) \\
 &= (\overline{N_{RTx}} - 1) \cdot t_{RTx}(N) + t_{ACK}(N)
 \end{aligned} \tag{4.37}$$

Considerando que la longitud de los datos pueda exceder la máxima capacidad de un paquete de 5 *slots*:

$$\overline{t_{ACK_{per}}}(N) = \begin{cases} 0 & N = 0 \\ (\overline{N_{RTx}} - 1) \cdot t_{RTx}(N) + t_{ACK}(N) & 0 < N \leq L_5 \\ ((\overline{N_{RTx}} - 1) \cdot t_{RTx}(L_5) + t_{ACK}(L_5)) \cdot \left\lfloor \frac{N}{L_5} \right\rfloor + \\ + \overline{t_{ACK_{per}}}(N \bmod L_5) & N > L_5 \end{cases} \tag{4.38}$$

El tiempo transcurrido desde que se inicia la transmisión de un paquete DM o DH hasta que se realiza la retransmisión, notado por $t_{RTx}(N)$, se ha de encontrar en el intervalo:

$$t_{RTx_{min}} \leq t_{RTx} \leq t_{RTx_{max}} \tag{4.39}$$

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

$$t_{RTx_{min}}(N) = \begin{cases} 0 & N = 0 \\ 2 \cdot T_S & 0 < N \leq L_1 \\ 4 \cdot T_S & L_1 < N \leq L_3 \\ 6 \cdot T_S & L_3 < N \leq L_5 \end{cases} \quad (4.40)$$

$$t_{RTx_{max}}(N) = T_{poll} \quad (4.41)$$

Puede observarse que la fórmula de $t_{RTx_{max}}(N)$, al igual que la ecuación 4.19 que define $t_{ACK_{poll}}(N)$, tiene en cuenta que los fragmentos intermedios de una PDU de L2CAP podrían estar separados hasta un máximo de n_{poll} slots. La figura 4.15 ilustra el significado de $t_{RTx_{max}}(N)$ para cada uno de los sentidos de comunicación, en el caso de la retransmisión de un paquete de 5 slots.

Sin embargo, el extremo inferior del intervalo, $t_{RTx_{min}}(N)$, definido por la ecuación 4.40, considera, tal y como se observa en la figura 4.16 que, para el sentido de transmisión esclavo-maestro (4.16 (b)), el maestro solicita la retransmisión inmediatamente y que, en el sentido maestro-esclavo (4.16 (a)), una vez que el maestro recibe la confirmación negativa del esclavo, procede a la retransmisión en el siguiente slot.

Finalmente, siguiendo el procedimiento indicado para $\overline{t_{ACK_{per}}}(N)$, la definición del tiempo medio de transmisión, $\overline{t_{TX_{per}}}(N)$, considerando las retransmisio-

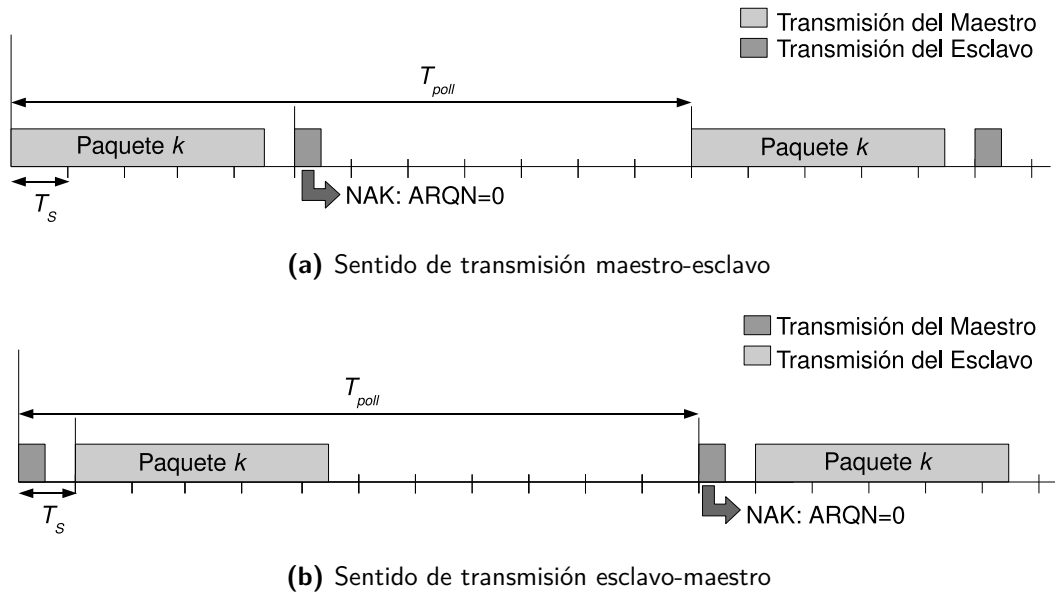


Figura 4.15: Tiempo transcurrido desde que se inicia la transmisión de un paquete de 5 *slots* hasta que se realiza la retransmisión, particularizando $t_{RTx}(N)$ al valor máximo, $t_{RTx_{max}}(N)$

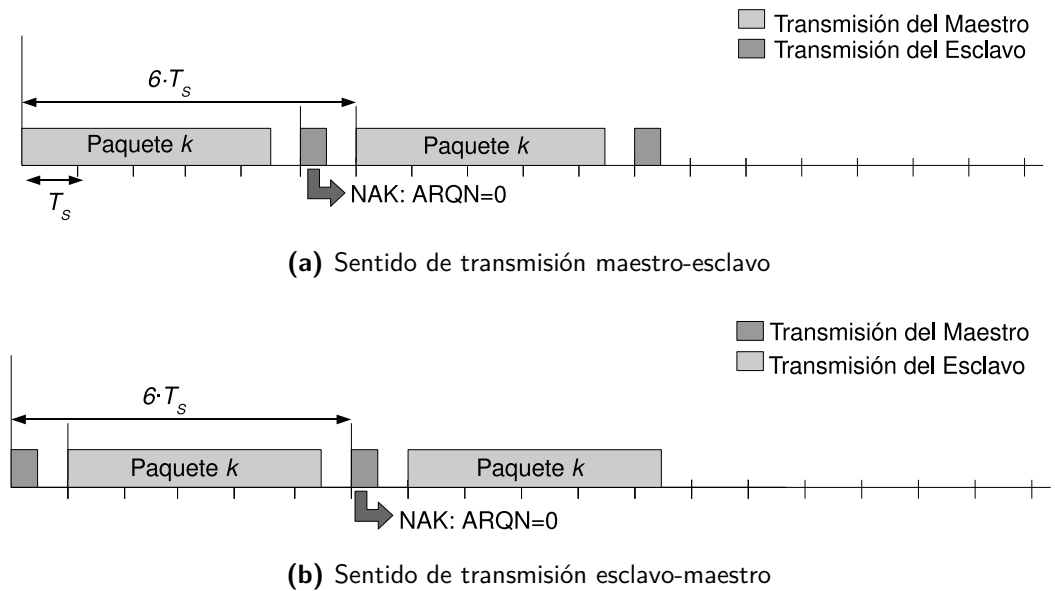


Figura 4.16: Tiempo transcurrido desde que se inicia la transmisión de un paquete de 5 *slots* hasta que se realiza la retransmisión, particularizando $t_{RTx}(N)$ al valor mínimo, $t_{RTx_{min}}(N)$

nes realizadas, responde a la siguiente ecuación:

$$\overline{t_{TX_{per}}}(N) = \begin{cases} 0 & N = 0 \\ (\overline{N_{RTx}} - 1) \cdot t_{RTx}(N) + n_b(N) \cdot t_b & 0 < N \leq L_5 \\ \overline{t_{ACK_{per}}}(L_5) \cdot \left\lfloor \frac{N}{L_5} \right\rfloor + \\ + \overline{t_{TX_{per}}}(N) \cdot (N \bmod L_5) & N > L_5 \end{cases} \quad (4.42)$$

Nótese que $\overline{t_{TX_{per}}}(N)$ tiene en cuenta las retransmisiones de los paquetes que transportan segmentos intermedios porque, al igual que en el caso ideal, no se va a transmitir el siguiente fragmento hasta no haber recibido la confirmación del actual.

Por consiguiente, la inclusión de 4.38 y 4.42 en la ecuación de retardo 4.21 permite definir $\overline{t_{RM_{per}}}(N)$, una cota máxima del retardo medio para el perfil SPP en el sentido de transmisión maestro-esclavo, teniendo en cuenta las retransmisiones:

$$\begin{aligned} \overline{t_{RM_{per}}}(N) = & T_{poll} + n_{fnf_R} \cdot \overline{t_{ACK_{per}}}(L_R + O_R(L_R) + C_L) + \\ & + \overline{t_{TX_{per}}}(L_{ff_R} + O_R(L_{ff_R}) + C_L) \end{aligned} \quad (4.43)$$

En el sentido de comunicación esclavo-maestro la cota $\overline{t_{RE_{per}}}(N)$ vendrá dada por la siguiente expresión:

$$\overline{t_{RE_{per}}}(N) = T_S + \overline{t_{RM_{per}}}(N) \quad (4.44)$$

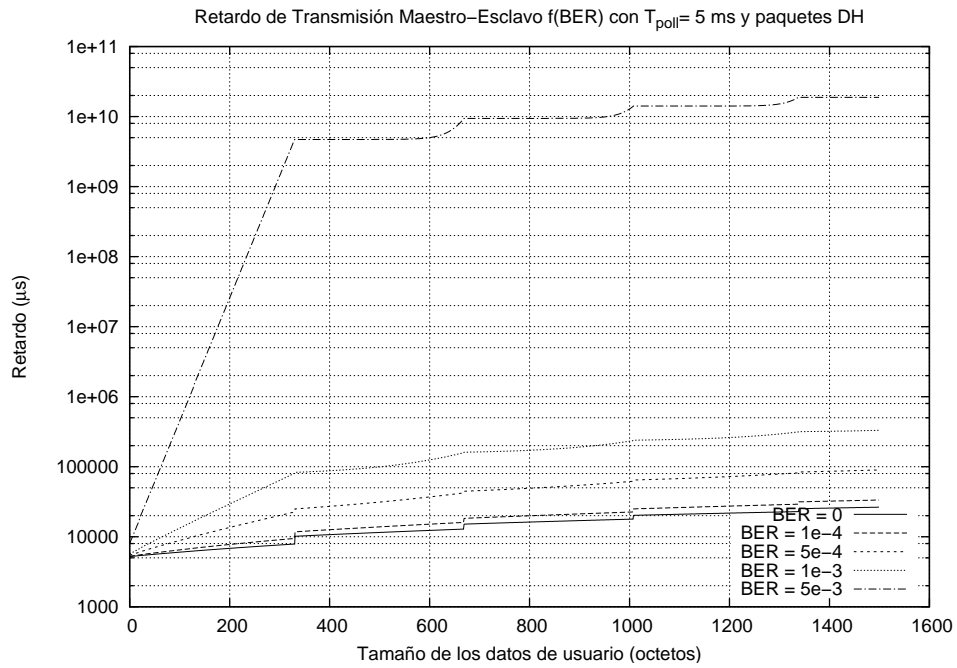
En la formula anterior, el *slot* añadido, T_S , respecto al caso del maestro, corresponde al *slot* anterior al inicio de la transmisión en el que el esclavo es direccionado por el maestro, como ya se explicó en la sección 4.2.

Es conveniente aclarar que para que las ecuaciones 4.43 y 4.44 sean cotas máximas del retardo medio se ha de particularizar $t_{RTx}(N)$ al valor máximo, $t_{RTx_{max}}(N)$, y $t_{ACK}(N)$ a $t_{ACK_{poll}}(N)$, definido, este último, por la ecuación 4.19.

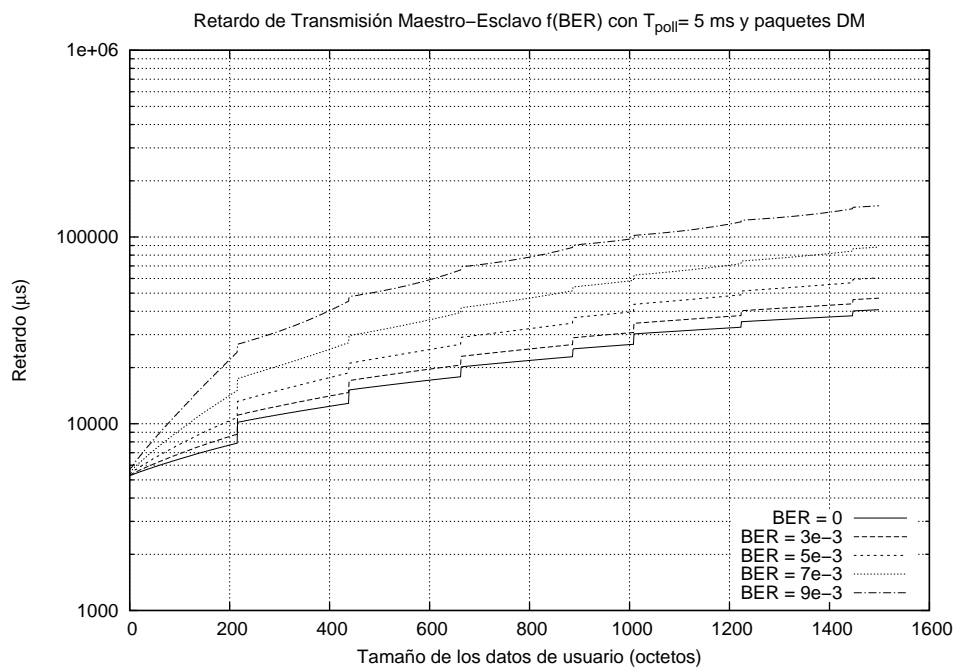
Las figuras 4.17 y 4.18 representan gráficamente los retardos máximos calculados para el perfil SPP, de acuerdo con las ecuaciones 4.43 y 4.44 aplicadas con distintos valores de tasa BER y un valor de T_{poll} de 5 ms, utilizando tanto paquetes DH como DM, en los sentidos de transmisión maestro-esclavo y esclavo-maestro, respectivamente.

Nótese que con tasas BER en torno al 0,1% ($BER = 10^{-3}$) el incremento del retardo puede ser excesivo para servicios en tiempo real, ya que se llega a alcanzar incluso 100 ms para tamaños de datos elevados. De hecho, para paquetes DH, una tasa BER del 0,5% ($BER = 5 \cdot 10^{-3}$) ya provoca una degradación inadmisibile. Sin embargo, el comportamiento con tasas BER cercanas al 1% utilizando paquetes DM (con protección contra errores) es aún aceptable.

4.3 Perfil SPP: Efecto de las retransmisiones en el retardo de transmisión

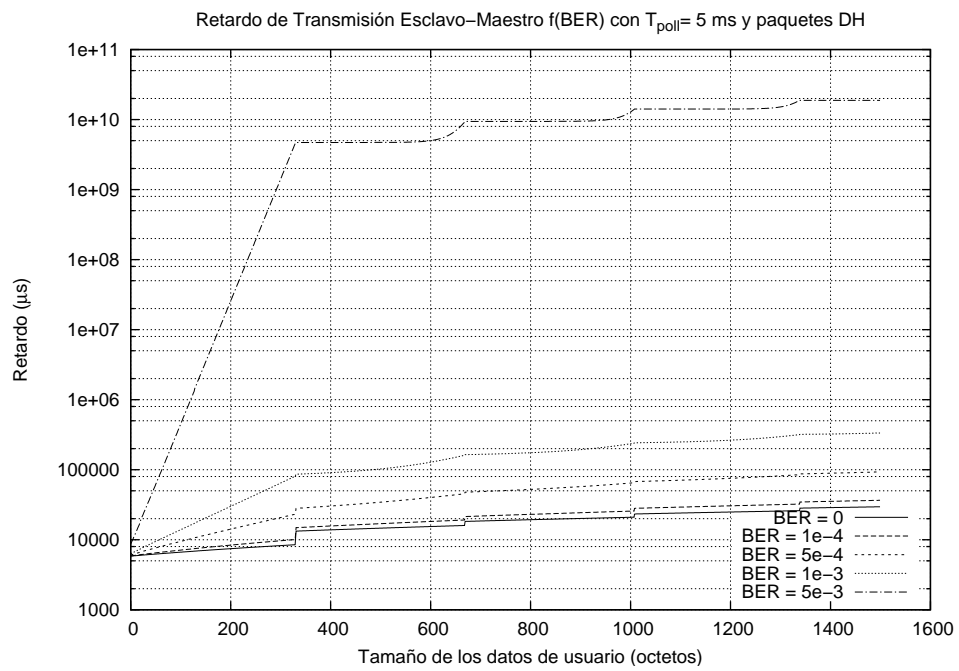


(a) Paquetes DH

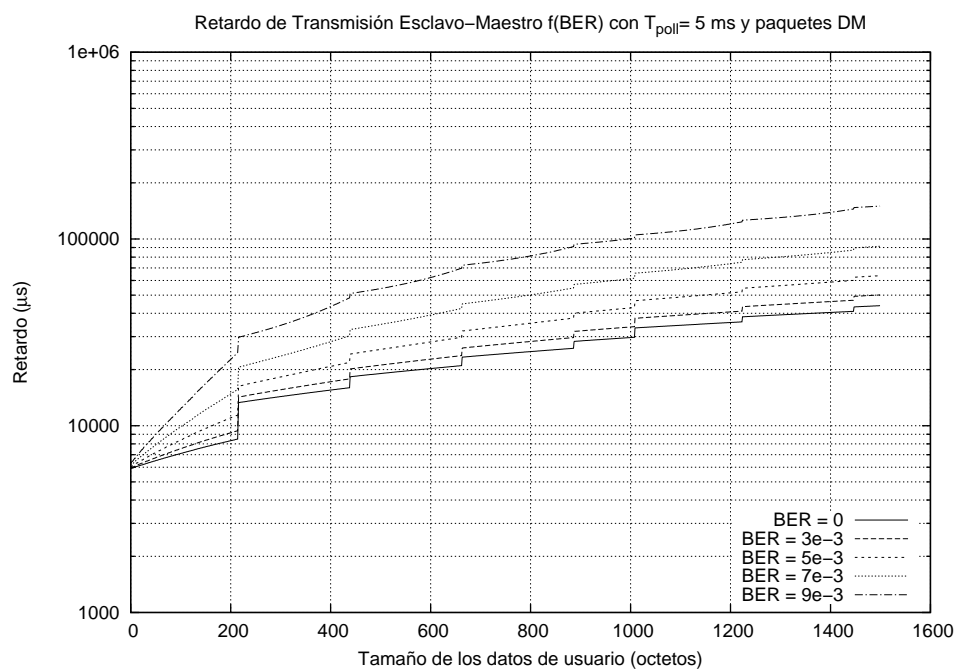


(b) Paquetes DM

Figura 4.17: Efecto de la tasa BER en las transmisiones maestro-esclavo para el perfil SPP



(a) Paquetes DH



(b) Paquetes DM

Figura 4.18: Efecto de la tasa BER en las transmisiones esclavo-maestro para el perfil SPP

Capítulo 5

Evaluación de los modelos mediante caracterización empírica

Las medidas del retardo que se describen en las siguientes secciones se han realizado con, al menos, dos dispositivos Bluetooth, con interfaz USB, conectados a un mismo PC con Sistema Operativo Ubuntu Linux 7.04 [Ubuntu Linux] y *kernel* de las versiones 2.6.12 y 2.6.20-*lowlatency*.

Se ha elegido Linux como Sistema Operativo por el hecho de que integra una implementación de código abierto del *Host* de la arquitectura de protocolos Bluetooth (véase la figura 2.1): BlueZ [BlueZ06].

BlueZ incluye, además de los niveles HCI *Firmware* y L2CAP, los protocolos RFCOMM y BNEP, y proporciona un controlador de interfaz de red virtual para poder establecer una red PAN. También dispone de utilidades de configuración de dispositivos Bluetooth y de una herramienta de monitorización de la comunicación entre el *Host* y el *Bluetooth Controller*.

Adicionalmente, BlueZ proporciona 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.

Como dispositivos Bluetooth se han empleado módulos USB de distintos fabricantes: Conceptronic [Conceptronic], Linksys [Linksys], D-Link [D-Link], Sitecom [Sitecom] que, en todos los casos, incorporan circuitos integrados del fabricante CSR (*Cambridge Silicon Radio*) [CSR] de la serie *BlueCore 2*, conformes a la especificación Bluetooth 1.1 y con las siguientes versiones y revisiones del *firmware* HCI y del nivel LM (obtenidas, precisamente, con un comando facilitado por BlueZ):

- *Versión de HCI: 1.1; Revisión de HCI: 0x20d y 0x33c*
- *Versión del nivel LM: 1.1; subversión del nivel LM: 0x20d y 0x33c*

Para los casos en los que el objetivo es evaluar el retardo evitando la interferencia, los módulos Bluetooth se han introducido en una caja metálica, tal y como se ilustra en la figura 5.1, que esquematiza la arquitectura *hardware* del sistema de medidas.

Una vez conocidos los recursos *hardware* y *software* empleados, en la sección 5.1 se pasa a describir la metodología de medida seguida para la evaluación de los modelos teóricos desarrollados en el capítulo 4. Posteriormente, en la sección 5.2 se contrasta la caracterización de los perfiles SPP y PAN, realizada analíticamente en el apartado 4.1.5, con los resultados experimentales, anticipando la influencia de la configuración de la calidad de servicio sobre

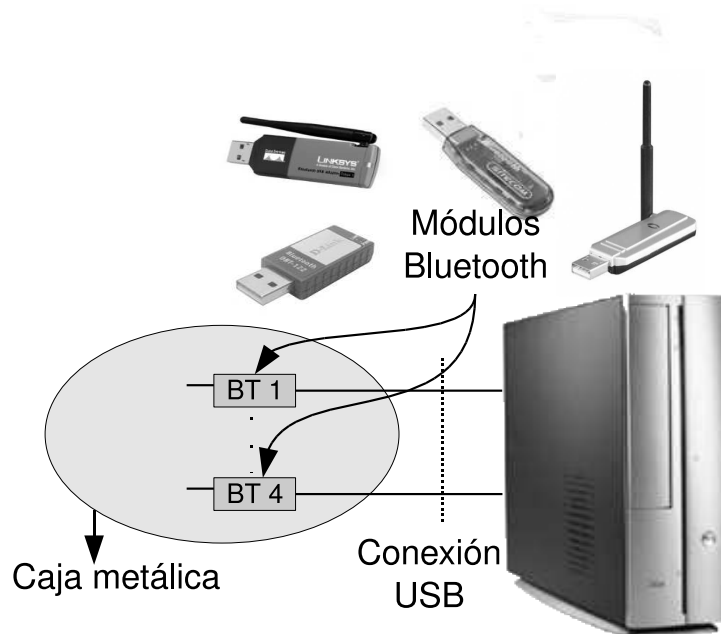


Figura 5.1: Esquema general de pruebas empleado para la caracterización del retardo de transmisión Bluetooth, utilizado para evitar las interferencias

el retardo, que se estudia exhaustivamente, para cada sentido de transmisión, en la sección 5.3. En esta sección también se lleva a cabo la corrección del modelo teórico descrito en la sección 4.2, teniendo en cuenta el comportamiento observado experimentalmente. Finalmente, la modificación del modelo teórico para contemplar el efecto de las retransmisiones, expuesta en la sección 4.3, es contrastada empíricamente en la sección 5.4.

5.1. Metodología de medida

En las pruebas realizadas se ha utilizado una arquitectura cliente-servidor programada en lenguaje C sobre la pila de protocolos BlueZ, en la que uno de los extremos, maestro o esclavo, transmite datos periódicamente, y el otro mide

el tiempo transcurrido desde el inicio de la transmisión de los datos hasta el instante de recepción. Por consiguiente, con el fin de garantizar la sincronización de los relojes de los extremos implicados en la comunicación, las pruebas se han ejecutado conectando las interfaces Bluetooth a un mismo PC, como se muestra en la figura 5.1.

Conviene precisar que, debido a que el retardo obviamente depende de que el tipo de paquete sea DM o DH, las pruebas se han efectuado fijando el uso de un tipo de paquete u otro. Para ello, previamente a la ejecución del *software* de medida, se configuran los tipos de paquete (DM o DH) que se permiten emplear, utilizando un comando de BlueZ.

Concretamente, el procedimiento de medida ha consistido en ejecutar, por cada conexión Bluetooth que requiriese la prueba, dos programas (escritos en lenguaje C) sobre el PC: un servidor y un cliente. Al programa servidor se le especifica como argumentos: el modo de funcionamiento (transmisión o recepción), el rango de tamaños a barrer a lo largo de la batería de pruebas y la periodicidad con la que se quiere transmitir (es decir, el tiempo que transcurre entre transmisiones consecutivas).

Tras establecerse la conexión entre servidor y cliente, el servidor transmite al cliente la configuración. Una vez que el cliente confirma al servidor la configuración, el transmisor (cliente o servidor, según el caso) envía, a intervalos fijos, 1000 paquetes por cada uno de los tamaños comprendidos entre las longitudes mínima y máxima que son, respectivamente, 10 y 1500 octetos. Este rango de longitudes se ha recorrido con incrementos de 10 octetos. Se decidió fijar el número de repeticiones realizado por cada longitud a 1000, con

la finalidad de que el tamaño muestral fuera lo bastante grande como para garantizar que la dispersión de los retardos medidos fuera baja y la media, representativa. Adicionalmente, se ha comprobado que duplicando el número de repeticiones y barriendo el rango de longitudes en incrementos unitarios no se consigue una mejora apreciable en la precisión de los resultados, y por contra, sí aumenta considerablemente el tiempo de ejecución de las pruebas.

El periodo de tiempo que están espaciadas cada una de las 1000 transmisiones es suficientemente elevado como para minimizar el efecto de encolamiento en los *buffers* del módulo *hardware*. La determinación del periodo de transmisión también requirió la realización de experimentos preliminares, que permitieron comprobar que con valores próximos a 10 ms se producían anomalías en las mediciones que no se presentaban con un orden de magnitud más.

Como ya se ha indicado anteriormente, en el extremo de recepción se ha medido el tiempo transcurrido desde que se inicia la transmisión de los datos hasta que se reciben. Para ello, como parte de los datos, se incluye la marca del instante de tiempo en el que se generó. En concreto, a lo largo de cada prueba, el extremo de recepción crea un fichero para cada longitud, en el que se almacena, para cada repetición, la diferencia entre el instante en el que se reciben los datos y el instante en el que se realizó la transmisión (esta información temporal es transportada como parte de los datos). Posteriormente, 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.

Finalmente, es conveniente aclarar que el *software* empleado para caracte-

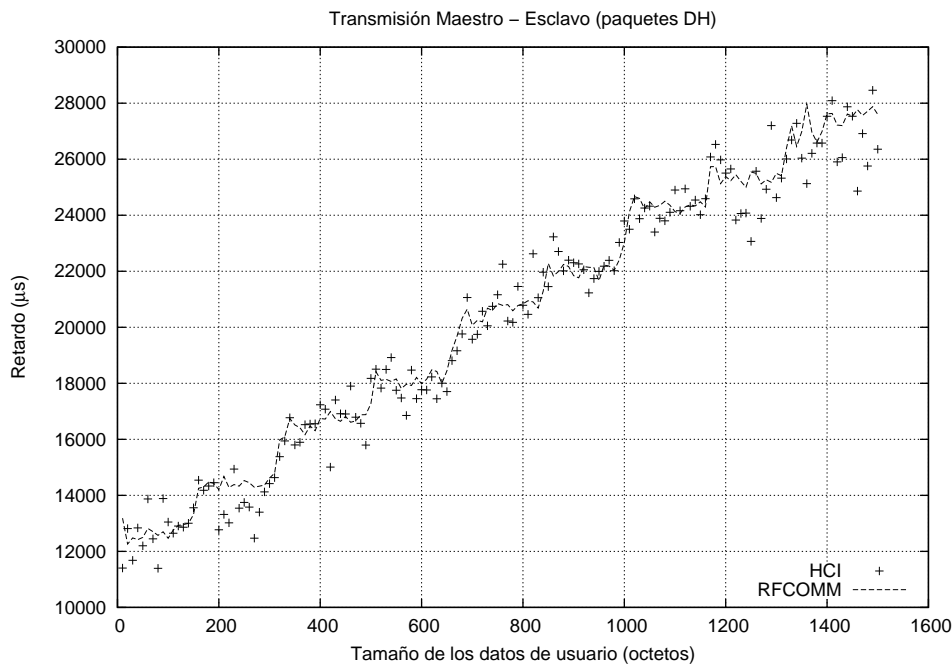


Figura 5.2: Comparación de los retardos de transmisión maestro-esclavo obtenidos a nivel RFCOMM y a nivel HCI, utilizando paquetes DH

rizar el retardo de la transmisión de datos y la implementación de los niveles RFCOMM, BNEP y L2CAP, no introducen distorsión alguna en el procedimiento de medida. Con el objeto de dejar constancia de este hecho, en la figura 5.2 se muestran las curvas de retardo obtenidas siguiendo el método descrito y, también, las resultantes del procesamiento de los ficheros de trazas generados en los extremos de transmisión y recepción por la herramienta de monitorización proporcionada por BlueZ. En estos ficheros se almacena el instante temporal en el que cada comando HCI o paquete de datos ACL va a ser enviado desde el *HCI Driver* al *HCI Firmware*, así como el instante en el que cada evento HCI o paquete de datos ACL es entregado por el *HCI Firmware* al *HCI Driver*, a través de la capa de transporte HCTL (véase la figura 2.1).

Nótese que la medida realizada a nivel HCI presenta una mayor dispersión.

Esto es debido a que ha sido promediada con tan solo 40 repeticiones, frente a las 1000 utilizadas en el caso de la caracterización llevada a cabo a nivel RFCOMM. La razón de utilizar un número tan reducido de muestras a nivel HCI es el tamaño excesivo de los ficheros generados por la herramienta de monitorización en cada uno de los extremos. Por ejemplo, para 12 longitudes distintas con 1000 repeticiones se requiere procesar dos ficheros de trazas de 67,7 MiB y de 59 MiB. Por este motivo y dada la inexistencia de diferencias entre los resultados, se optó por realizar las mediciones a nivel de RFCOMM.

5.2. Comparación del retardo teórico y medido: Perfiles SPP y PAN

Las medidas del retardo para los perfiles SPP y PAN se han realizado para una *piconet* de dos dispositivos, en la que, como se ha indicado anteriormente, los dos módulos Bluetooth se han introducido en una caja metálica para evitar el efecto de posibles interferencias.

En las pruebas realizadas se ha empleado una arquitectura cliente-servidor específica para el perfil PAN, que no utiliza directamente la interfaz de *sockets* Bluetooth, sino una interfaz de *sockets* TCP/UDP convencional que utiliza la interfaz de red virtual proporcionada por BlueZ para una red PAN. Por este motivo, en el caso del perfil PAN, la ejecución del servidor y del cliente va precedida de la ejecución de un *script* para el establecimiento de la red PAN y de la asignación de interfaces de red virtuales.

Una vez ejecutados los programas servidor y cliente, en el perfil SPP, tras establecerse la conexión RFCOMM, el servidor transmite al cliente la configuración de la prueba. En el caso del perfil PAN, esta configuración se realiza utilizando una conexión TCP que se cerrará tras su finalización. Seguidamente, el extremo que vaya a realizar las transmisiones abrirá un *socket* UDP, a través del que se transmitirán los datos.

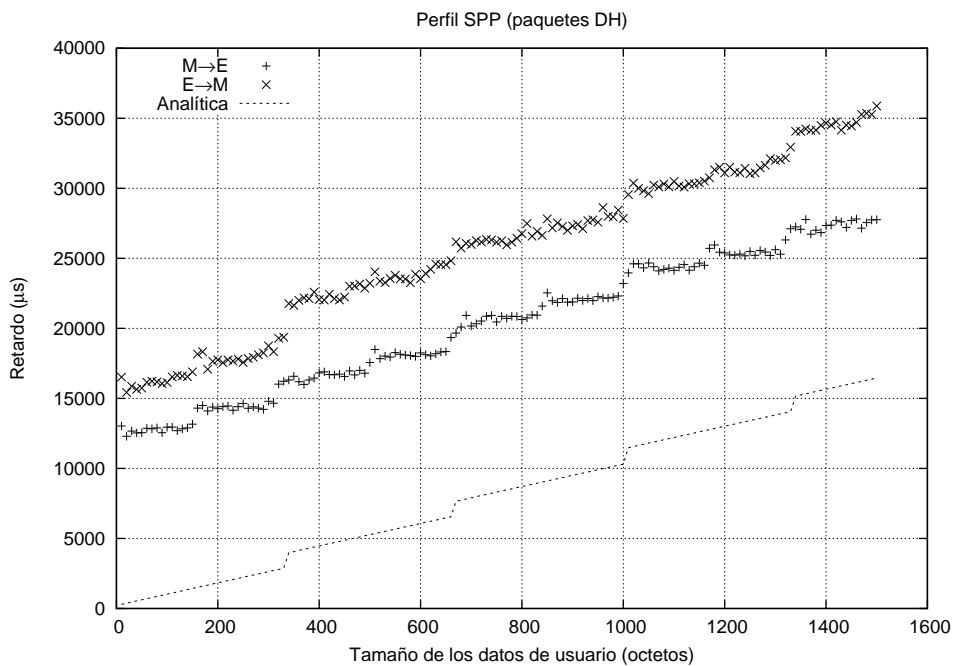
La medida se ha efectuado para ambos perfiles siguiendo la metodología descrita en la sección 5.1 (con las salvedades indicadas) en los siguientes casos: 1) maestro transmitiendo paquetes DH, 2) maestro transmitiendo paquetes DM, 3) esclavo transmitiendo paquetes DH y 4) esclavo transmitiendo paquetes DM.

A continuación se presenta la comparación entre los retardos mínimos teóricos calculados cuando el maestro realiza la transmisión y las medias de los retardos medidos para el perfil SPP (figura 5.3¹) y para el perfil PAN (figura 5.4) en los casos en los que la transmisión es efectuada por el maestro y por el esclavo.

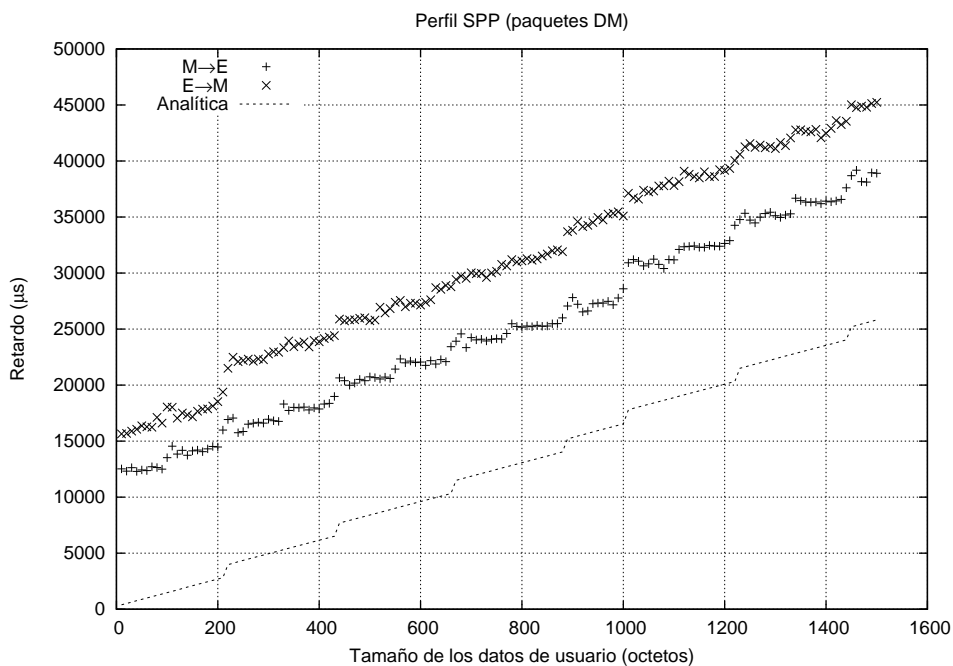
Puede observarse que las pendientes de crecimiento de los retardos medidos coinciden con la del mínimo teórico, y que los escalones que se producen en la gráfica del retardo suceden a las mismas longitudes en el caso teórico y en el experimental. Por ejemplo, si se analizan los paquetes DH, el primer salto se produce entre los 330 y los 340 octetos, valores para los que se pasa de utilizar un único paquete DH5 a emplear un paquete DH5 y un paquete DH1. En general, tanto para paquetes DH como para paquetes DM, los escalones tienen lugar cada vez que se requiere un paquete de 5 *slots* adicional.

¹En esta figura y en las sucesivas, las abreviaturas « $M \rightarrow E$ » y « $E \rightarrow M$ » denotan, respectivamente, el sentido de transmisión maestro-esclavo y esclavo-maestro.

5.2 Comparación del retardo teórico y medido: Perfiles SPP y PAN

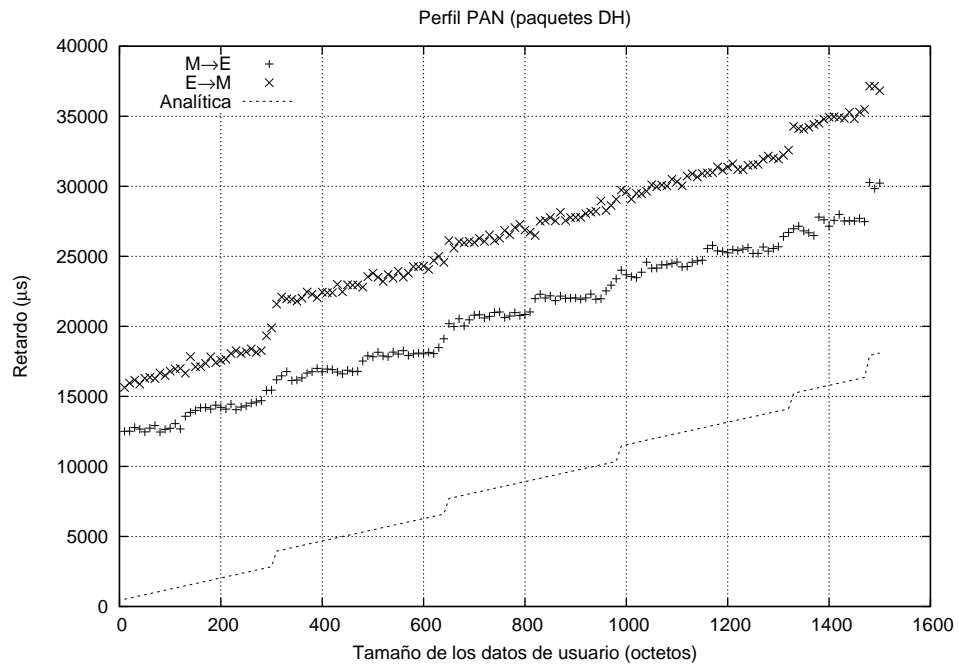


(a) Paquetes DH

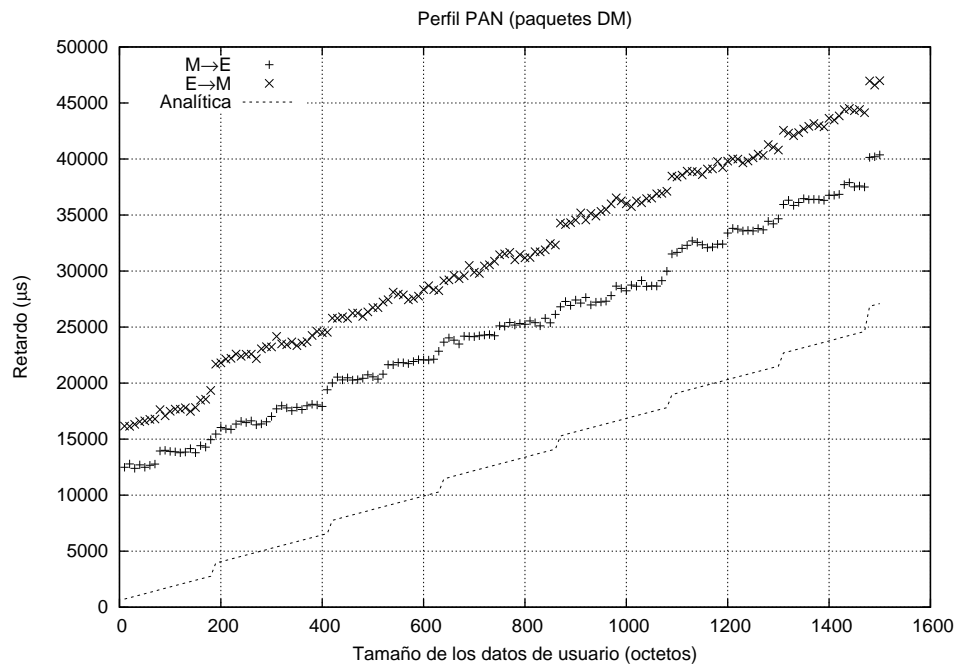


(b) Paquetes DM

Figura 5.3: Comparación del retardo medio de transmisión medido con el calculado analíticamente, mediante la ecuación 4.6, para el perfil SPP



(a) Paquetes DH



(b) Paquetes DM

Figura 5.4: Comparación del retardo medio de transmisión medido con el calculado analíticamente, mediante la ecuación 4.9, para el perfil PAN

Nótese que en el caso de SPP el salto es mayor cuando el tamaño de los datos de usuario es de 1010 octetos, es decir, cuando se supera el valor del parámetro $N1 = 1008$ octetos fijado en la implementación de BlueZ [BlueZ06]. La razón es que la transmisión de 1010 octetos requiere fragmentar los datos de usuario, siendo necesario para su transmisión el uso de dos tramas RFCOMM: la primera de 1008 octetos, y la segunda de 2 octetos, de forma que al sumarle el *overhead* correspondiente a los protocolos RFCOMM y L2CAP resultan, respectivamente, 1017 octetos y 10 octetos que, dependiendo de que se utilicen paquetes DH o DM, requerirán tantos paquetes como se indican en la tabla 5.1.

Tipo de paquete	1.º fragmento (1017 octetos con <i>overhead</i>)	2.º fragmento (10 octetos con <i>overhead</i>)
DH	3 paquetes DH5	1 paquete DH1
DM	4 paquetes DM5 y 1 paquete DM3	1 paquete DM1

Tabla 5.1: Paquetes Bluetooth necesarios para la transmisión de 1010 octetos de datos de usuario utilizando el perfil SPP

En las gráficas correspondientes al perfil PAN, el salto equivalente se produce cuando el número de octetos de usuario supera el valor de 1472 octetos, tal y como se indicó en el apartado 4.1.7. En este caso, la transmisión de una longitud de 1480 octetos de datos de usuario mediante UDP supondría que, en el nivel IP, los 1488 octetos (1480 octetos + 8 octetos de cabecera UDP) tendrían que fragmentarse en dos tramas para que, al agregarle la cabecera IP, no se superasen los 1500 octetos. Así, resultarían dos datagramas IP de 1500 y de 28 octetos de longitud que, al ser encapsuladas en tramas BNEP y posteriormente en PDUs de L2CAP, acabarían alcanzando un tamaño de 1507 octetos y de 35 octetos (tras sumar 3 y 4 octetos correspondientes a las cabeceras de los protocolos BNEP y

L2CAP), que serían transmitidas con los paquetes que se detallan en la tabla 5.2.

Tipo de paquete	1.º fragmento (1507 octetos con <i>overhead</i>)	2.º fragmento (35 octetos con <i>overhead</i>)
DH	4 paquetes DH5 y 1 paquete DH3	1 paquete DH3
DM	6 paquetes DM5 y 1 paquete DM3	1 paquete DM3

Tabla 5.2: Paquetes Bluetooth necesarios para la transmisión de 1480 octetos de datos de usuario utilizando el perfil PAN

Finalmente, cabe apuntar que el *offset* en el que se diferencia el retardo calculado teóricamente y el medido para el sentido de transmisión maestro-esclavo está relacionado, como se comprobará en la siguiente sección, con el valor del periodo T_{poll} . Obsérvese que el desplazamiento vertical respecto al teórico previsto es aproximadamente de 12,5 ms, valor que coincide con $T_{poll}/2$, ya que al no haber configurado la calidad de servicio, el valor por defecto del parámetro T_{poll} , definido en la especificación Bluetooth [Bluetooth SIG01c], es de 25 ms.

En lo que respecta al sentido de transmisión esclavo-maestro, puede apreciarse que el *offset* es superior a $T_{poll}/2$, en un valor de unos 2 ms. De este tiempo, 625 μ s corresponderían al *slot* en el que el esclavo es direccionado por el maestro, y el tiempo restante, 1,375 ms (realmente irrelevante para la práctica totalidad de los servicios), podría ser introducido, bien por el sistema operativo, bien por el propio dispositivo cuando actúa como esclavo, por razones de implementación específicas del fabricante.

5.3. Efecto de la configuración de la calidad de servicio

El estudio que se expone en esta sección se ha llevado a cabo para el perfil SPP configurando la calidad de servicio mediante el comando *HCI_QoS_Setup* (véase el apartado 4.6.6 de la parte H:1 de la especificación [Bluetooth SIG01c]). Este comando permite solicitar al nivel LM los parámetros de calidad de servicio relacionados con la especificación de flujo del nivel L2CAP descritos en el apartado 2.1.5.

El comando *HCI_QoS_Setup* es utilizado por el nivel L2CAP de ambos extremos durante la transacción de configuración de calidad de servicio. Sin embargo, debido a que para el perfil SPP no se especifica que el nivel L2CAP deba proporcionar una determinada calidad de servicio distinta de la establecida por defecto, tras el establecimiento de la conexión RFCOMM se ha procedido, mediante el comando indicado, a configurar los parámetros de calidad. Para ello, se ha tenido en cuenta que, en las especificaciones del fabricante CSR [CSR], se explica que el tipo de servicio proporcionado es *Best Effort* y que el parámetro *Latency*, indicado en el comando *HCI_QoS_Setup*, es convertido de microsegundos a *slots* y utilizado directamente para intentar fijar a ese valor el periodo T_{poll} . Si la negociación de la calidad de servicio que se lleva a cabo entre entidades homólogas del nivel LM tiene éxito, en la respuesta al comando *HCI_QoS_Setup* que se envía al *Host* el parámetro *Latency* coincidirá con el valor de T_{poll} expresado en microsegundos.

Partiendo del conocimiento del tipo de servicio implementado y del parámetro que afecta al periodo T_{poll} , se ha comenzado caracterizando, de forma exhaustiva, el retardo de las conexiones punto a punto en cada uno de los sentidos de transmisión para una *piconet* de dos dispositivos y distintos valores del parámetro *Latency*.

Los resultados de las medidas de retardo obtenidas para los sentidos de transmisión maestro-esclavo y esclavo-maestro en la *piconet* de dos dispositivos se presentan en el apartado 5.3.1, donde también se contrasta el comportamiento observado experimentalmente con el previsto teóricamente en la sección 4.2. Posteriormente, en el apartado 5.3.2, para una *piconet* de tres dispositivos, se proponen cotas del retardo de transmisión teniendo en cuenta el incremento del número de esclavos y el valor de T_{poll} configurado. Además, las cotas propuestas se cotejan con los retardos medidos para cada una de las conexiones establecidas en la *piconet*. A continuación, en el apartado 5.3.3 se generalizan las cotas formuladas para una *piconet* de dos esclavos y, finalmente, se comparan gráficamente con los resultados obtenidos al incorporar a la *piconet* un tercer esclavo (apartado 5.3.4).

En todos los casos (*piconets* de 2, 3 y 4 dispositivos), se ha utilizado también una arquitectura cliente-servidor por cada conexión Bluetooth, y se ha medido en los extremos de recepción, al igual que antes, el tiempo transcurrido desde que se inició la transmisión de los datos de usuario hasta que son recibidos. Para ello, justo antes del envío por el *socket* se incluye, como parte de los datos, la hora proporcionada por el sistema.

Tal y como se indicó en la sección 5.1, para cada uno de los tamaños de los datos de usuario, se ha promediado el retardo obtenido para 1000 transmisiones, igualmente espaciadas un periodo de 100 ms.

De nuevo, con el fin de evitar problemas de sincronización, las medidas se han efectuado sobre un mismo PC con interfaces USB Bluetooth.

5.3.1. Medidas para una *piconet* de dos dispositivos

Como ya se ha anticipado, las medidas del retardo para evaluar el efecto de la calidad de servicio se han llevado a cabo, para SPP, configurándola con los valores del parámetro *Latency* especificados en la tabla 5.3, en la que también se indican los intervalos de *poll* correspondientes.

<i>Latency</i>	Intervalo de <i>poll</i> (n_{poll})	<i>Latency</i>	Intervalo de <i>poll</i> (n_{poll})
1250 μs	2 <i>slots</i>	15000 μs	24 <i>slots</i>
2500 μs	4 <i>slots</i>	20000 μs	32 <i>slots</i>
5000 μs	8 <i>slots</i>	25000 μs	40 <i>slots</i>
10000 μs	16 <i>slots</i>		

Tabla 5.3: Valores del parámetro *Latency* (y del intervalo de *poll* al que equivale) para los que se han realizado las medidas de evaluación del efecto de la configuración de la calidad de servicio

Los resultados de las medidas realizadas en cada uno de los sentidos de transmisión, maestro-esclavo y esclavo-maestro, se exponen, respectivamente, en los apartados 5.3.1.1 y 5.3.1.2.

En ambos casos se comienza presentando el retardo obtenido con paquetes DH y valores pequeños del parámetro *Latency*, con el fin de determinar el retardo

mínimo que puede conseguirse. Seguidamente, se presentan los resultados obtenidos para los restantes valores del intervalo de *poll*, utilizando paquetes DH y DM, y se muestra que el hecho de no configurar la calidad de servicio es equivalente a fijar el parámetro *Latency* a un valor de 25 ms.

A continuación, en el apartado 5.3.1.3 se comparan entre sí ambos sentidos de transmisión. Finalmente, en los apartados 5.3.1.4 y 5.3.1.5, se contrasta y corrige el modelo analítico para cada sentido de comunicación.

5.3.1.1. Sentido de transmisión maestro-esclavo

Las gráficas mostradas en la figura 5.5 ponen de manifiesto que el uso de valores del parámetro *Latency* inferiores a 10 ms conduce a curvas de retardo prácticamente indistinguibles de la obtenida para 10 ms.

Como posible fundamento cabría argumentar que cuando se transmiten datos utilizando un paquete de 5 *slots* no es posible garantizar un intervalo de *poll* de 1,25 ms (2 *slots*). En ese caso, la coincidencia entre los retardos, tendría lugar para aquellos tamaños de datos en los que se requiriese el uso de paquetes *multislot*. Por consiguiente, dado que la coincidencia se produce independientemente de la longitud, se descarta que ésta sea la razón.

Otra hipótesis, que no puede comprobarse, es que la utilización de una arquitectura estructurada de acuerdo con la figura 2.1 que, como se explicó en la sección 2.1, incluye una interfaz de comunicación física HCI, imponga esta limitación.

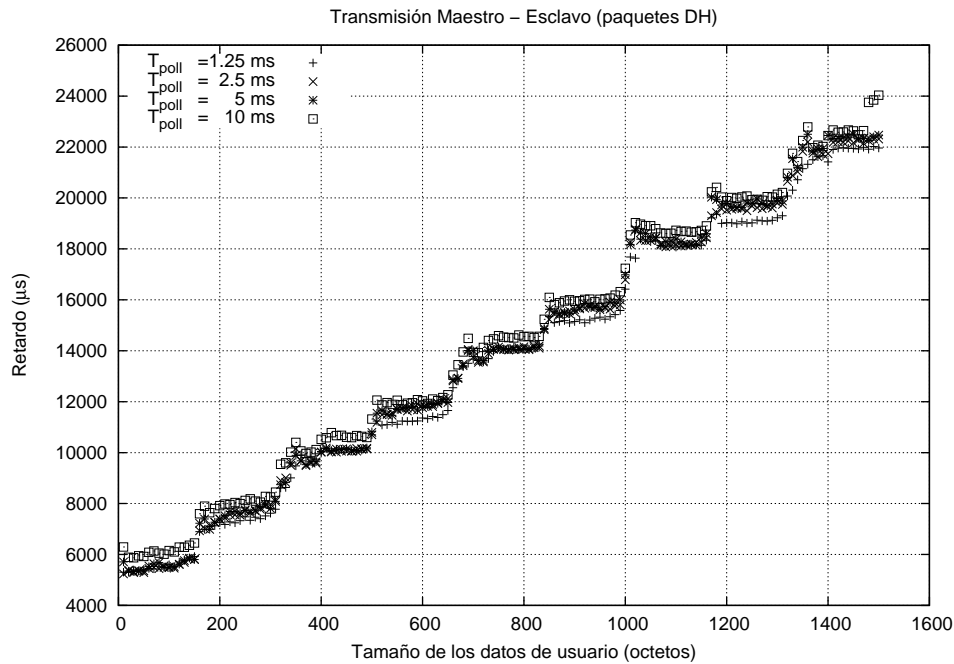
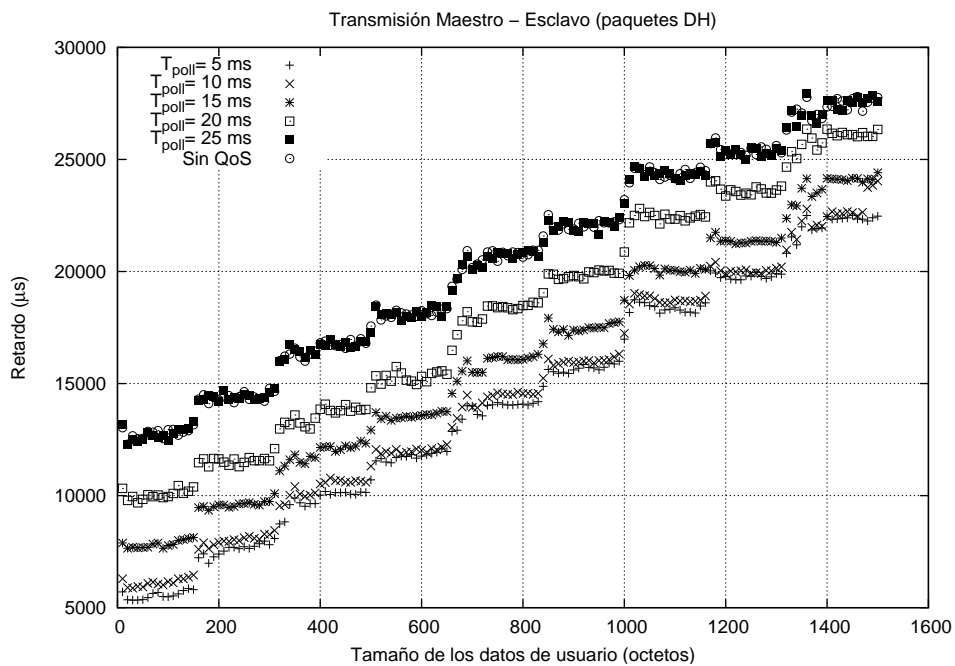


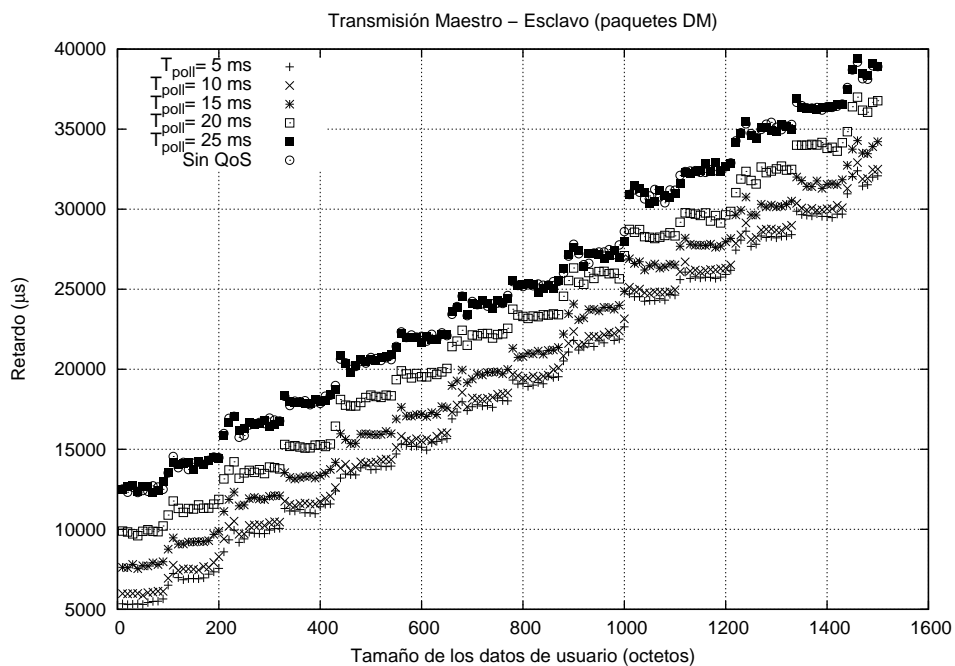
Figura 5.5: Comparación del retardo medio de transmisión maestro-esclavo para valores de T_{poll} (asimilable al parámetro *Latency*) inferiores a 15 ms

No obstante, independientemente de la causa de esta restricción, puede concluirse que aunque el valor mínimo del intervalo de *poll* sea de 1,25 ms (véase el apartado 2.1.3), a efectos prácticos, el mínimo se encuentra en los 10 ms (16 *slots*), al menos, para el sentido de transmisión maestro-esclavo.

En la figura 5.6 se representan los retardos medidos para SPP cuando la transmisión es efectuada por el maestro, con paquetes DH y paquetes DM, sin configurar la calidad de servicio y configurándola para los valores del parámetro *Latency* indicados en la tabla 5.3 a partir de 5 ms. En las gráficas representadas en esta figura, puede observarse que, cuando no se ha configurado la calidad de servicio, el retardo coincide con el obtenido para un intervalo de *poll*, n_{poll} , de 40 *slots*, (o T_{poll} de 25 ms) que es el valor establecido por defecto en la especificación Bluetooth [Bluetooth SIG01c], tal y como se anticipó al final de la



(a) Paquetes DH



(b) Paquetes DM

Figura 5.6: Comparación del retardo medio de transmisión maestro-esclavo para distintos valores del parámetro T_{poll}

sección 5.2 (compárese con la gráfica correspondiente al sentido maestro-esclavo de la figura 5.3, para la que no se ha fijado ningún valor del parámetro *Latency*).

Otro hecho relevante que se pone de manifiesto en la figura 5.6 es que la altura de los escalones en la gráfica del retardo no depende, como se había previsto teóricamente, del valor de T_{poll} empleado. Esto es, cuanto más elevado es el valor de n_{poll} , y más paquetes de 5 *slots* se necesitan para transmitir los datos, mayores tendrían que ser los sucesivos escalones equiespaciados que corresponden al llenado de un paquete adicional de 5 *slots*. Sin embargo, no se produce este comportamiento modelado por la ecuación 4.21. Por consiguiente, aunque el maestro sólo esté obligado a comunicarse con cada esclavo de la *piconet* una vez por cada n_{poll} *slots*, en la realidad, el maestro, tras direccionar por primera vez al esclavo, continúa haciéndolo mientras tenga datos que enviarle. De esta forma, el intervalo de *poll* sólo es tenido en cuenta por el maestro una vez que deje de transmitir al esclavo, para saber cuándo tiene que volver a direccionarlo.

En definitiva, cuando el maestro no tiene más que un esclavo al que atender, el valor de T_{poll} no afecta al tiempo de separación entre envíos sucesivos de cada uno de los paquetes correspondientes a fragmentos intermedios de una PDU de L2CAP. De hecho, el efecto que se produce al modificar el parámetro T_{poll} es el de un desplazamiento vertical, esto es, un *offset* constante e independiente de la longitud de los datos a transmitir, pero claramente relacionado con el valor de T_{poll} . Este *offset* que, para valores del intervalo de *poll* superiores a los 5 ms es 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 sigue una distribución uniforme entre 0 y T_{poll} . Sin embargo, en el caso de un intervalo de *poll* de 5 ms o menos, el *offset* no es nunca inferior a 5 ms. Por este motivo, las gráficas de retardo medio obtenidas para T_{poll} de 5 y 10 ms son casi coincidentes, ya que el *offset* para el segundo caso es $T_{poll}/2$, es decir, 5 ms.

No obstante, aunque la inclusión de la ecuación 4.19 en las ecuaciones del retardo 4.6 y 4.10 permite obtener una cota máxima, es factible adaptar el modelo de retardo para la transmisión maestro-esclavo con el fin de que se refleje el comportamiento real de los dispositivos (véase el apartado 5.3.1.4).

5.3.1.2. Sentido de transmisión esclavo-maestro

La figura 5.7 ilustra que, al igual que en el sentido contrario, emplear valores del parámetro *Latency* inferiores a 5 ms no supone alcanzar un retardo menor. Por ejemplo, la figura muestra que con un intervalo de *poll* de 1,25 ms, el retardo para tamaños de datos que pueden ser transmitidos en un único paquete Bluetooth es mayor que el alcanzado con un T_{poll} de 5 ms. Además, la curva de retardo correspondiente a un T_{poll} de 2,5 ms coincide, o incluso, supera la de T_{poll} de 5 ms, una vez que se excede la capacidad de un paquete DH5.

Cabe resaltar que, en este caso, las gráficas de retardo correspondientes a T_{poll} de 5 ms y de 10 ms, sí se diferencian más que cuando es el maestro el que transmite. Por tanto, no parece razonable configurar un valor de parámetro *Latency* que se encuentre por debajo del mínimo, que en esta ocasión parece ser de 5 ms, en vez de 10 ms, como ocurría en el sentido de transmisión maestro-esclavo.

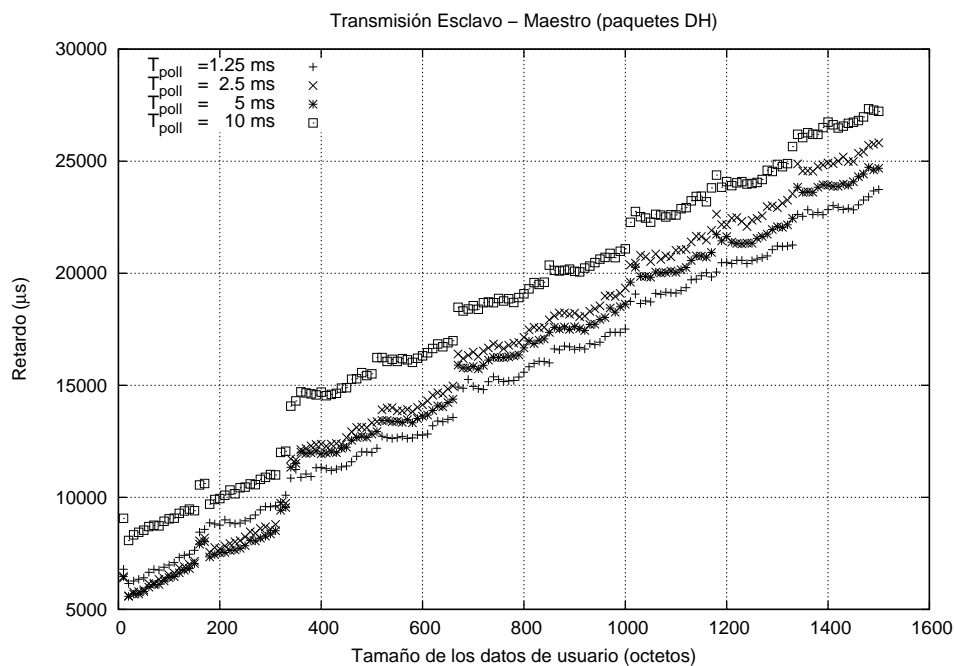
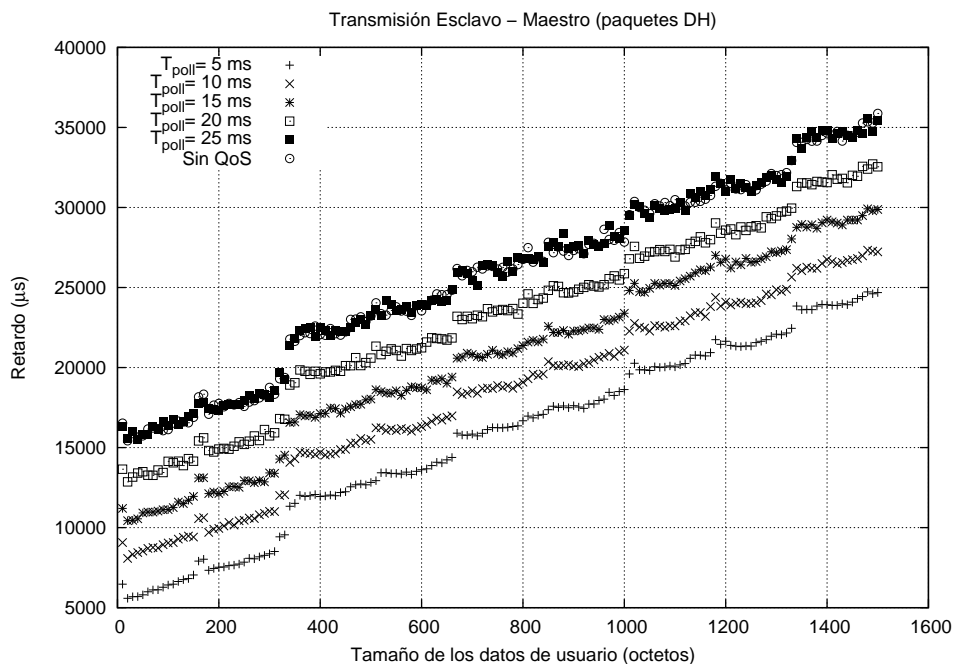


Figura 5.7: Comparación del retardo medio de transmisión esclavo-maestro para valores del parámetro T_{poll} inferiores a 15 ms

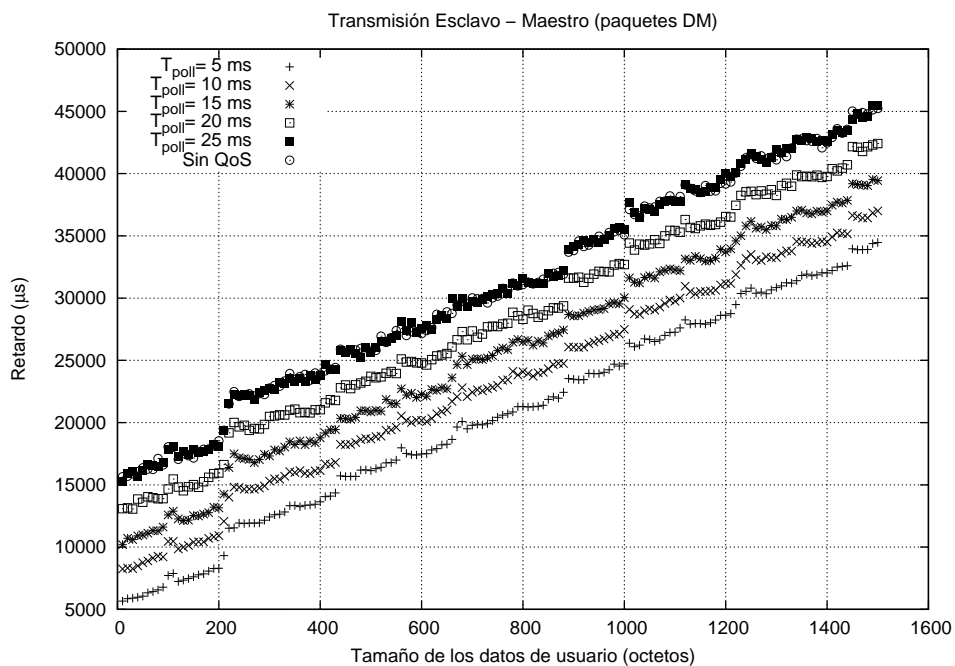
De hecho, para valores del parámetro *Latency* de 5 ms, o superiores, no se produce el comportamiento observado en la figura 5.7, tal y como se aprecia en la figura 5.8, en la que se muestran los retardos medidos en las mismas condiciones descritas en el apartado anterior cuando las transmisiones se efectúan en el sentido esclavo-maestro.

Al igual que en el sentido de transmisión maestro-esclavo, cuando no se ha configurado la calidad de forma explícita (fijando un valor del parámetro *Latency* mediante el comando *HCI_QoS_Setup*), el retardo coincide con el obtenido para un intervalo de *poll*, n_{poll} , de 40 slots, que es el valor establecido por defecto.

También en este caso se concluye que el valor de T_{poll} no afecta al tiempo de confirmación de cada uno de los paquetes correspondientes a fragmentos intermedios de una PDU de L2CAP. Una vez más, el efecto que se produce al



(a) Paquetes DH



(b) Paquetes DM

Figura 5.8: Comparación del retardo medio de transmisión esclavo-maestro para distintos valores del parámetro T_{poll}

modificar el parámetro T_{poll} es el de un desplazamiento vertical, es decir, un *offset* constante, independiente de la longitud de los datos a transmitir, y también, al igual que en el sentido de transmisión maestro-esclavo, relacionado con el intervalo de *poll*. Como se verá en el apartado 5.3.1.5, el *offset* es superior, en aproximadamente 2 ms, al indicado para la transmisión del maestro ($T_{poll}/2$), al que hay que sumar el *slot* de direccionamiento. El origen de estos 2 ms, en principio desconocido, podría atribuirse a una distinta gestión de la política de atención a las interrupciones asociadas a las peticiones de envío de datos, dependiendo de que la unidad Bluetooth esté actuando como maestro o esclavo.

La adaptación del modelo de retardo para la transmisión esclavo-maestro, necesaria para reflejar el comportamiento observado en las medidas, se realiza en el apartado 5.3.1.5.

5.3.1.3. Comparación de los retardos de transmisión en los sentidos maestro-esclavo y esclavo-maestro

En las figuras 5.9, 5.10, 5.11, 5.12 y 5.13, se comparan los resultados de los dos apartados anteriores: el retardo de transmisión medido en el sentido maestro-esclavo y el medido para la comunicación esclavo-maestro. La comparación se ha realizado para los valores del parámetro *Latency* indicados en la tabla 5.3 superiores a 2500 μ s.

Puede observarse que los escalones que se producen en el retardo tras el llenado de un paquete de cinco *slots*, en el caso de que el envío lo esté realizando el maestro, tienen siempre la misma altura hasta que se excede la MTU de L2CAP, que para RFCOMM es de 1013 octetos.

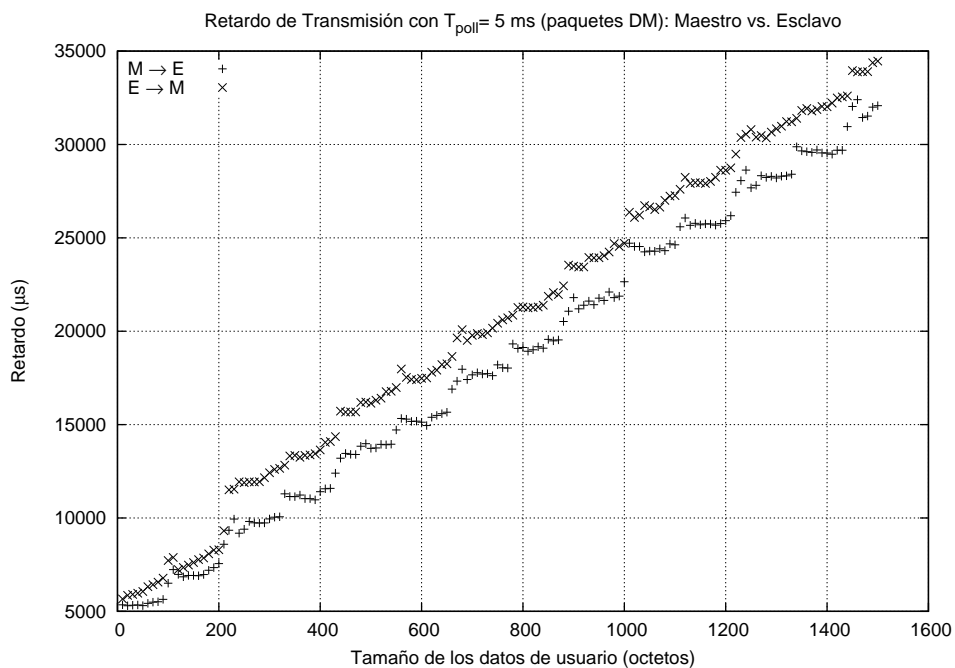
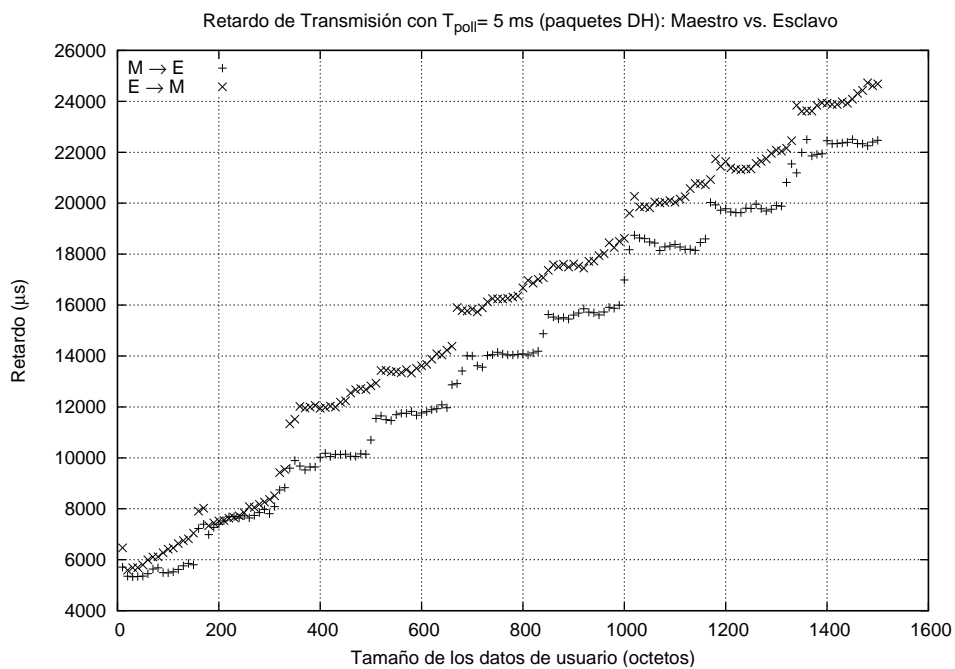
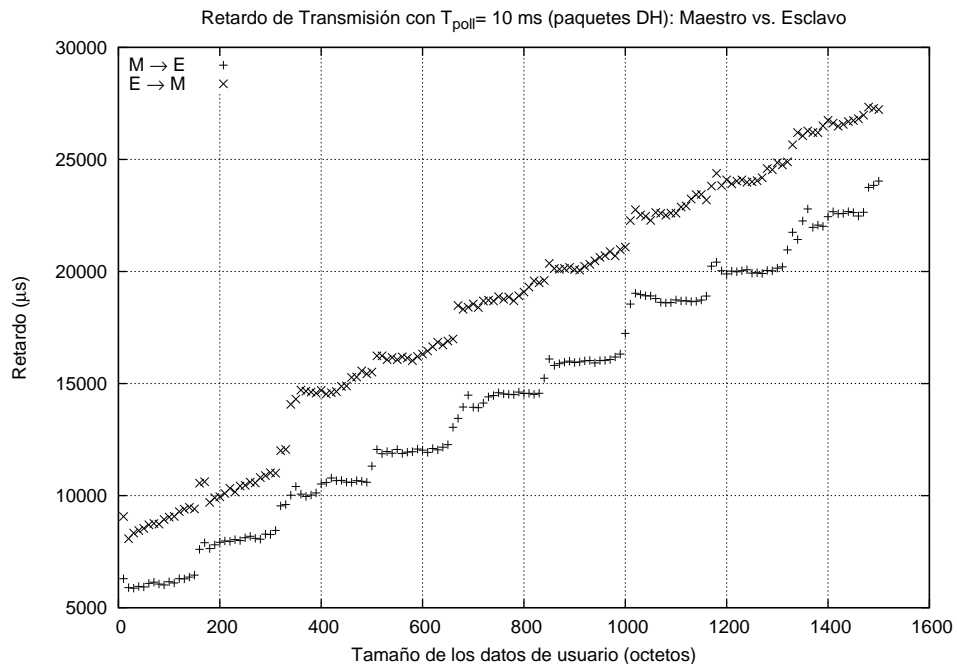
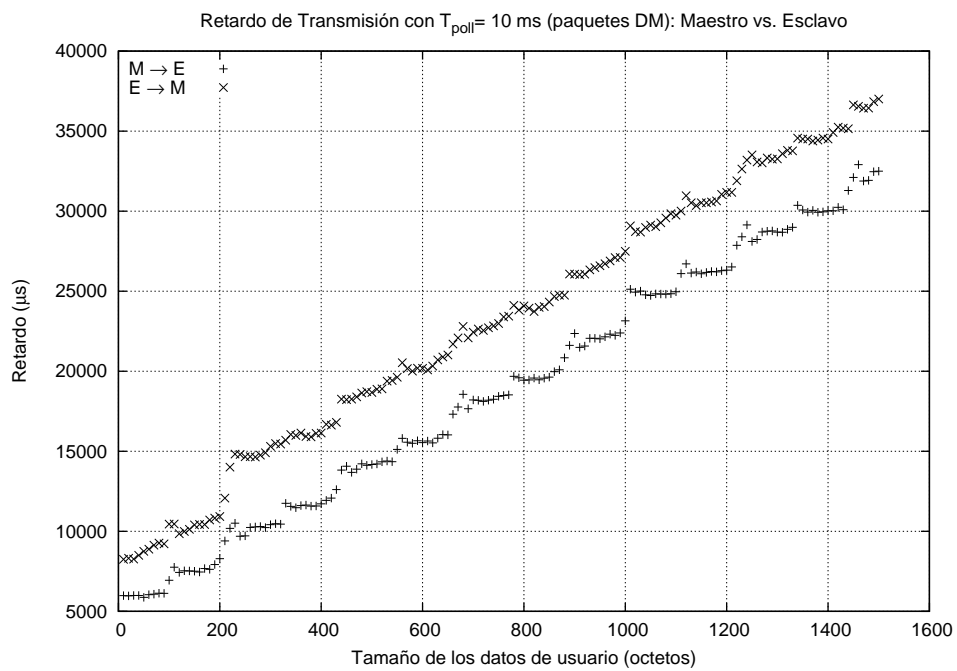


Figura 5.9: Comparación de los retardos medios de transmisión en los sentidos maestro-esclavo y esclavo-maestro para un valor del parámetro T_{poll} de 5 ms

5.3 Efecto de la configuración de la calidad de servicio



(a) Paquetes DH



(b) Paquetes DM

Figura 5.10: Comparación de los retardos medios de transmisión en los sentidos maestro-esclavo y esclavo-maestro para un valor del parámetro T_{poll} de 10 ms

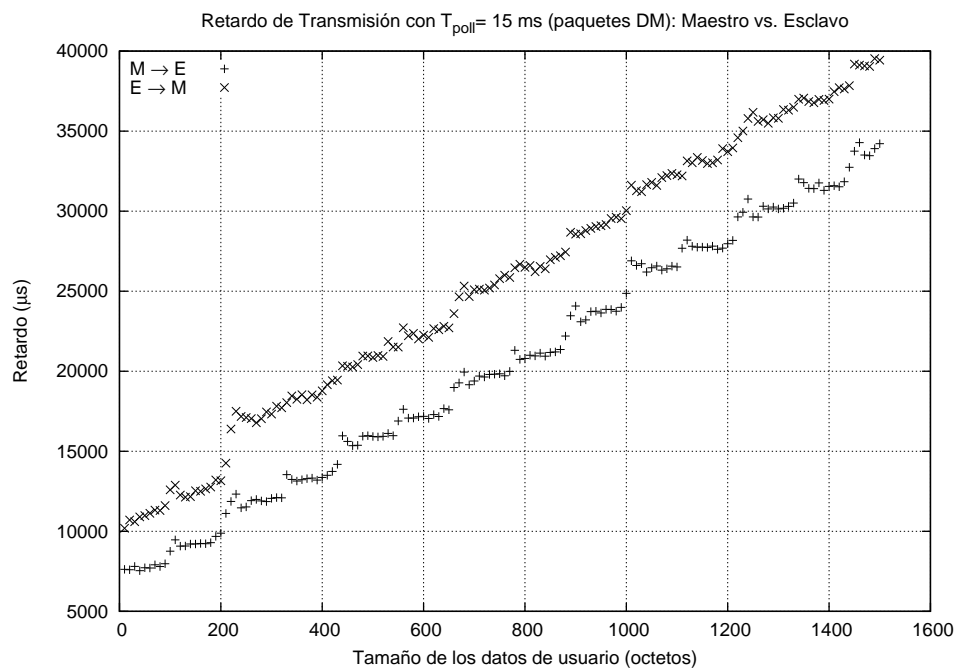
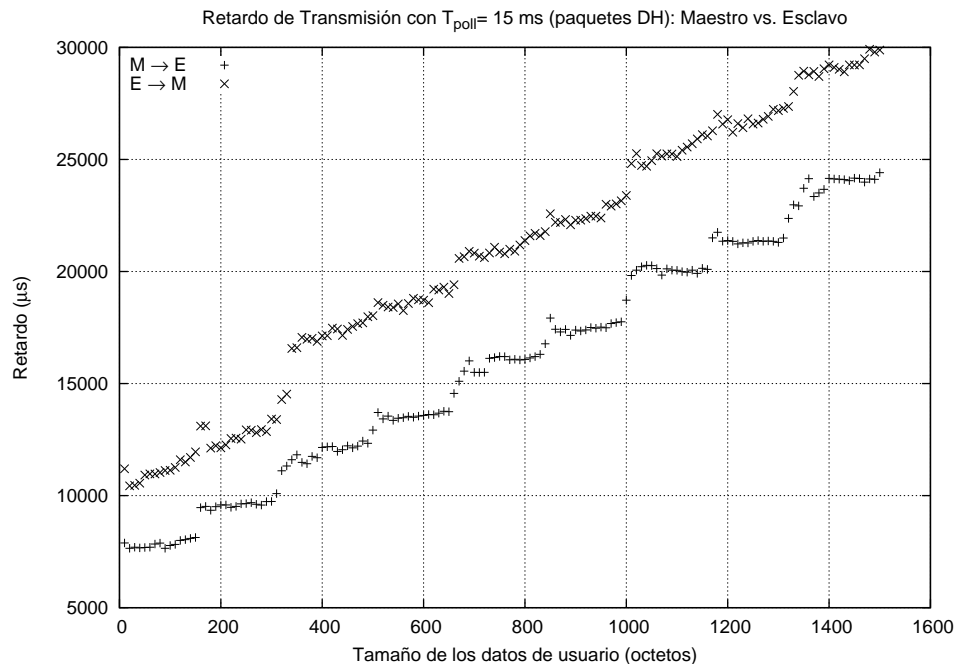
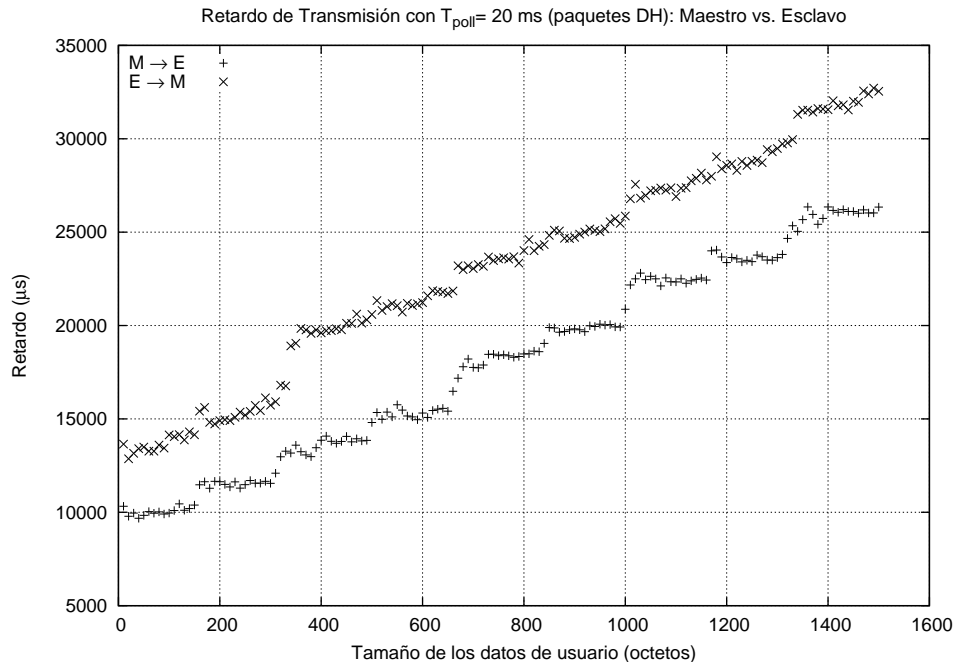
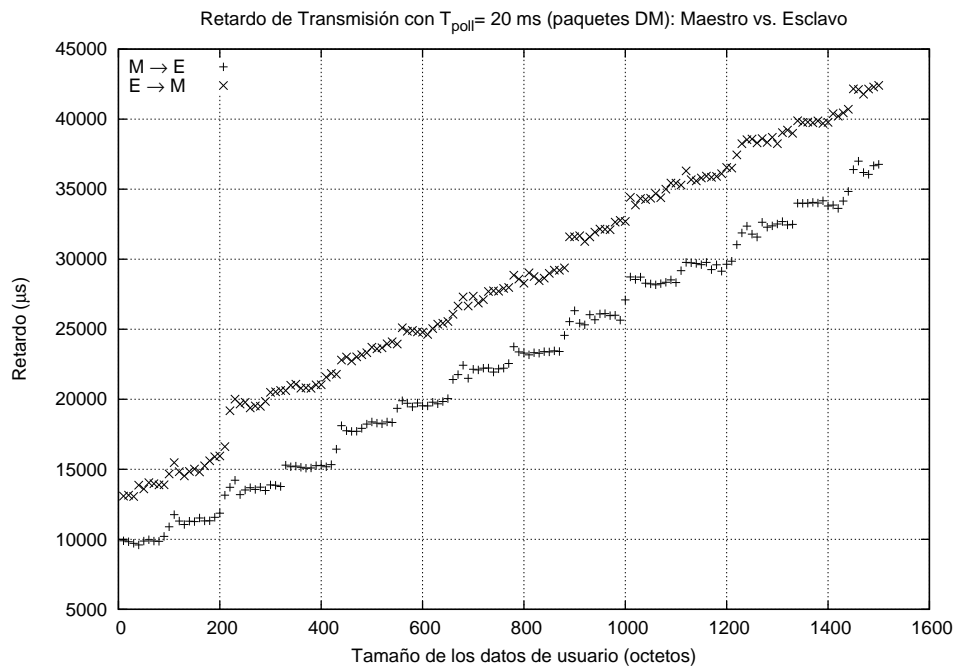


Figura 5.11: Comparación de los retardos medios de transmisión en los sentidos maestro-esclavo y esclavo-maestro para un valor del parámetro T_{poll} de 15 ms

5.3 Efecto de la configuración de la calidad de servicio



(a) Paquetes DH



(b) Paquetes DM

Figura 5.12: Comparación de los retardos medios de transmisión en los sentidos maestro-esclavo y esclavo-maestro para un valor del parámetro T_{poll} de 20 ms

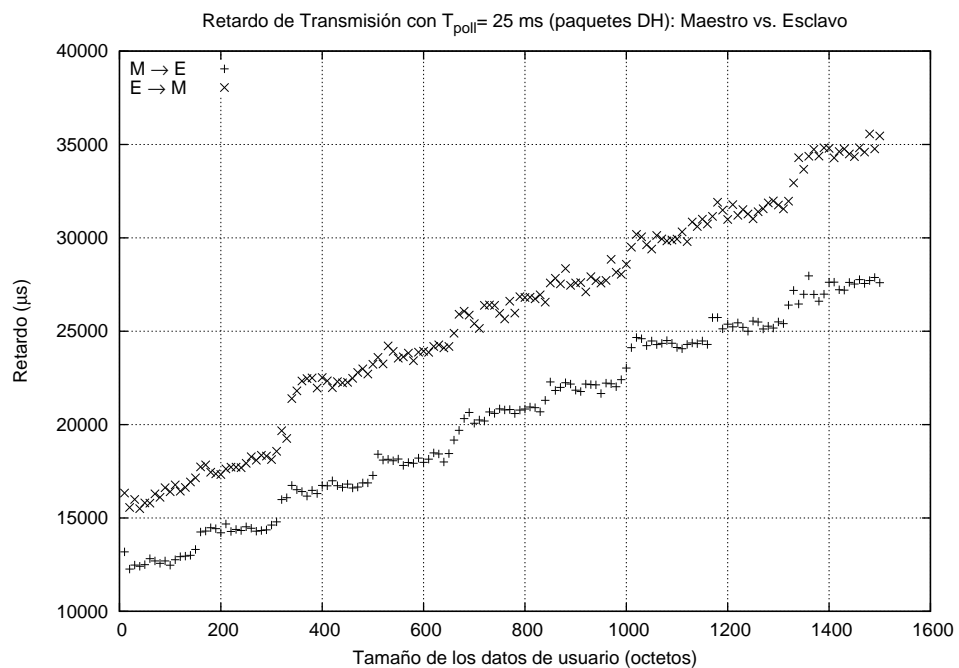
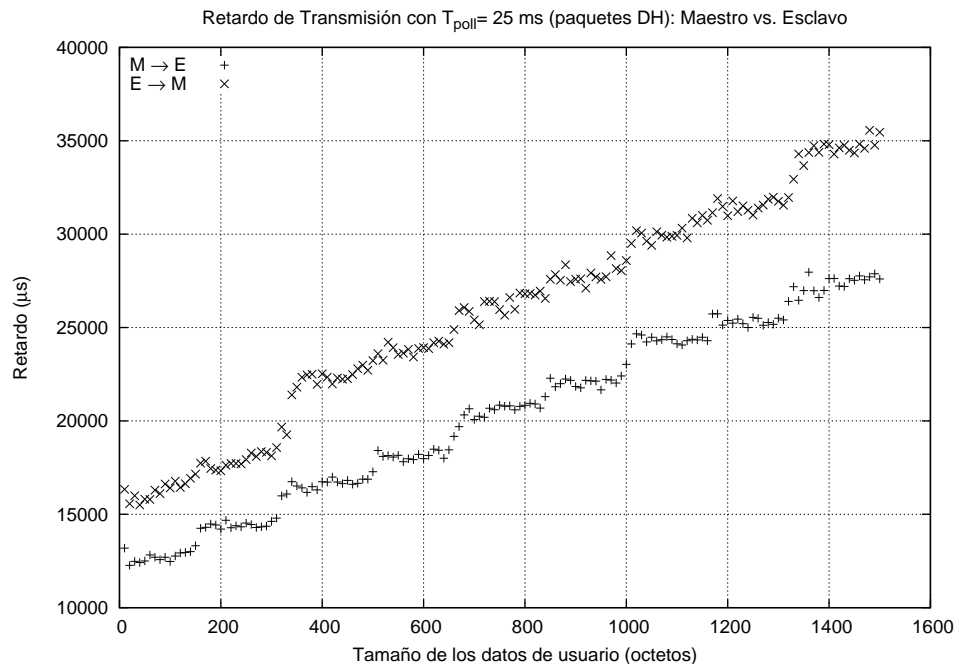


Figura 5.13: Comparación de los retardos medios de transmisión en los sentidos maestro-esclavo y esclavo-maestro para un valor del parámetro T_{poll} de 25 ms

Sin embargo, cuando es el esclavo el que está transmitiendo, el primer salto incremental que se produce en el retardo es notablemente mayor que los restantes. Dicho con otras palabras, el incremento en el retardo es mayor cuando el tamaño de la PDU de L2CAP, por exceder la capacidad de un paquete DH5 o 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*. Este primer salto es superior a 2 *slots*, en todos los casos, por lo que no parece depender del parámetro T_{poll} .

La hipótesis que se formula como justificación de este fenómeno es la siguiente: cuando el maestro recibe un paquete 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 PDU 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 PDU de L2CAP ($L_{CH} = 10$, véase el apartado 2.1.2.4), poniéndose de manifiesto este comportamiento cuando la PDU 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.

Obsérvese que se dice que la verificación se haría para el caso de que el paquete recibido contenga fragmentos iniciales. Esto quiere decir que tras la recepción de los sucesivos fragmentos de continuación, el maestro seguiría direccionando al esclavo mientras que éste no le respondiera con un paquete

NULL, tal y como se apuntó en el apartado 5.3.1.1. De no ser así, el incremento se produciría por cada fragmento de la PDU de L2CAP.

Adicionalmente y para reforzar la hipótesis planteada, cabe destacar que el primer escalón es también mayor que el que tiene lugar para la misma longitud cuando es el maestro el que transmite.

5.3.1.4. Corrección del retardo en la transmisión maestro-esclavo

Las ecuaciones que se propusieron en la sección 4.2 para tener en cuenta el efecto de la configuración de calidad en el retardo teórico han de ser modificadas atendiendo al resultado de las medidas realizadas.

La corrección propuesta consiste en considerar que la componente del retardo $t_{ACK}(N)$ no debe diferir de la definida en condiciones ideales por la ecuación 4.1, ya que, según se ha observado experimentalmente, el tiempo de separación entre envíos de paquetes Bluetooth que transportan segmentos intermedios de una PDU L2CAP no depende del valor de T_{poll} negociado.

Por consiguiente, el retardo de transmisión en el sentido maestro-esclavo para SPP, vendría dado, para el caso peor, por la siguiente ecuación:

$$t_{R_M}(N) = T_{poll} + t_R(N) \quad (5.1)$$

y para el caso medio:

$$\overline{t_{R_M}}(N) = (T_{poll}/2) + t_R(N) \quad (5.2)$$

donde, como se anticipó en el apartado 5.3.1.1, $T_{poll}/2$ es precisamente la media de una variable aleatoria que sigue una distribución uniforme entre 0 y T_{poll} .

En las figuras 5.14, 5.15, 5.16, 5.17 y 5.18 se comparan los retardos de transmisión obtenidos teóricamente mediante las ecuaciones 5.1 y 5.2 para los valores del parámetro *Latency* indicados en la tabla 5.3, a partir de $5000 \mu s$, con los medidos en el sentido maestro-esclavo (que ya se representaron en la figura 5.6 del apartado 5.3.1.1).

Recuérdese que, según se dijo en el apartado 5.3.1.1, para intervalos de *poll* superiores o iguales a 10 ms o más, la curvas de retardo obtenidas empíricamente presentaban un *offset* que coincidía con $T_{poll}/2$. Este hecho puede constatarse en las figuras 5.15–5.18, en las que se aprecia que el retardo medio, dado por la ecuación 5.2, se ajusta al retardo medido. Sin embargo, para un valor de T_{poll} de 5 ms, la coincidencia se produce con el caso peor, obtenido por la ecuación 5.1, debido a que el intervalo de *poll* real que implementa el dispositivo, con alta probabilidad, será superior.

Aunque la aproximación conseguida es bastante aceptable para todos los casos, en todas estas gráficas puede observarse que, por cada salto predicho teóricamente, en la práctica, se producen dos escalones. Este doble escalón parece desmentir el hecho de que cuando el esclavo recibe el último bit entrega el paquete a los niveles superiores. Por contra, los resultados empíricos atestiguan que el esclavo espera un tiempo fijo por paquete, independientemente del número exacto de bits del último fragmento, ya que tamaños de datos distintos, que para su transmisión requieren el mismo número de paquetes, sufren el mismo retardo.

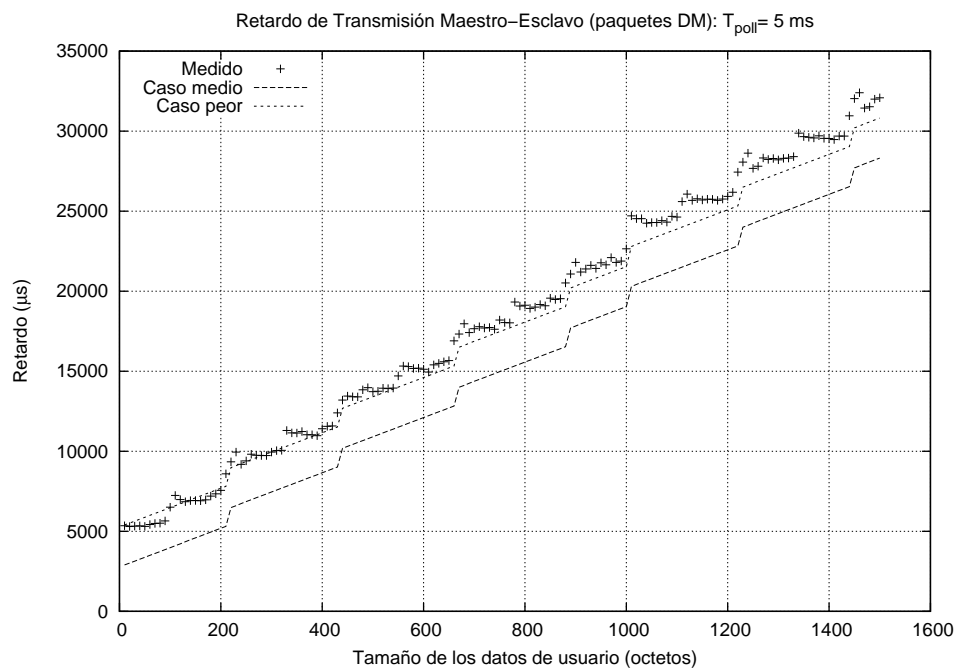
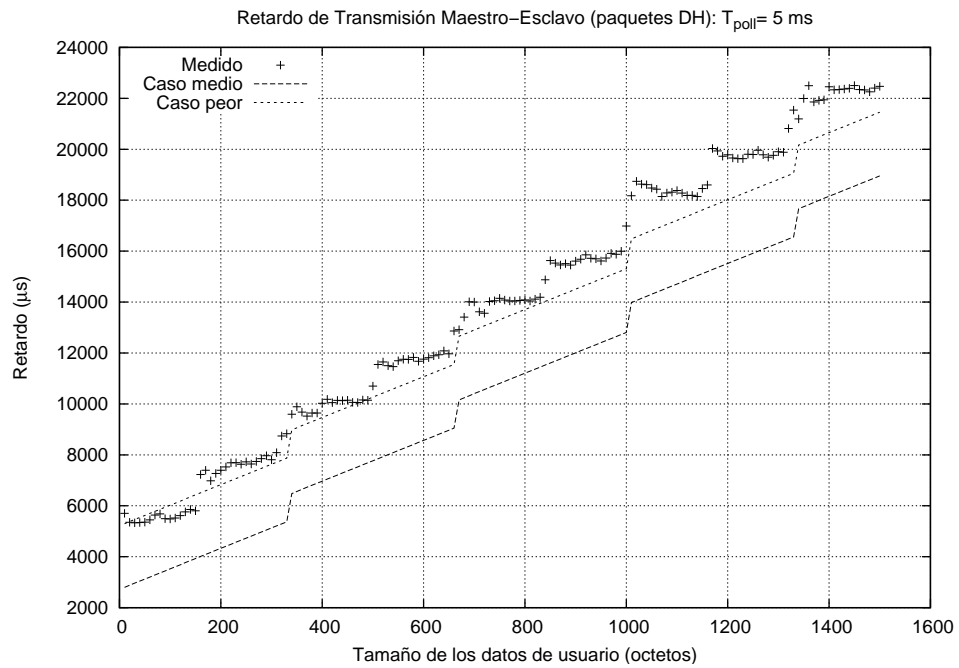
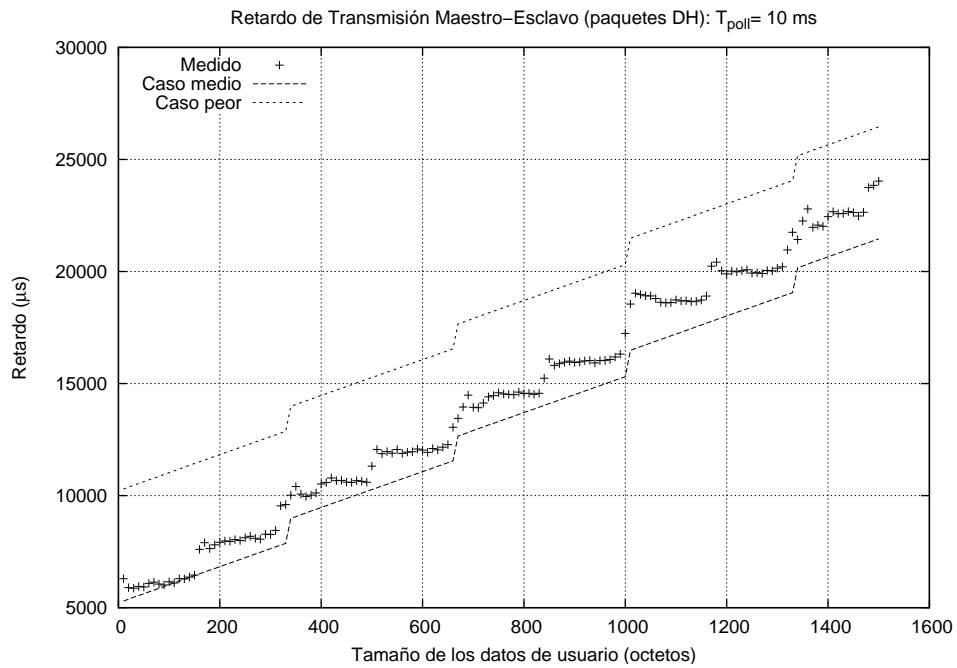
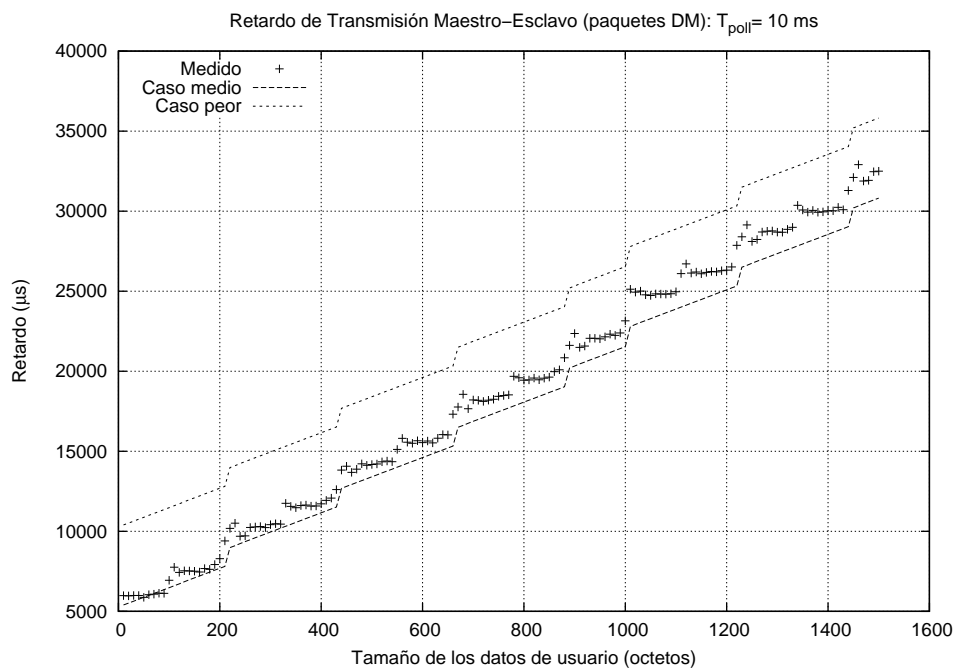


Figura 5.14: Comparación de los retardos de transmisión maestro-esclavo medido y teórico corregido para un valor del parámetro T_{poll} de 5 ms

5.3 Efecto de la configuración de la calidad de servicio

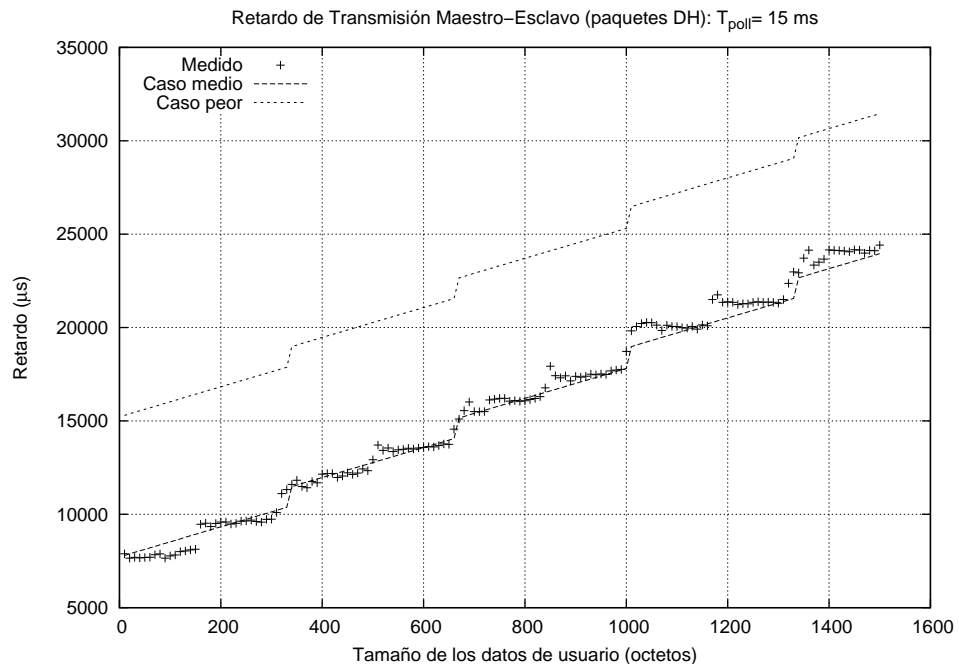


(a) Paquetes DH

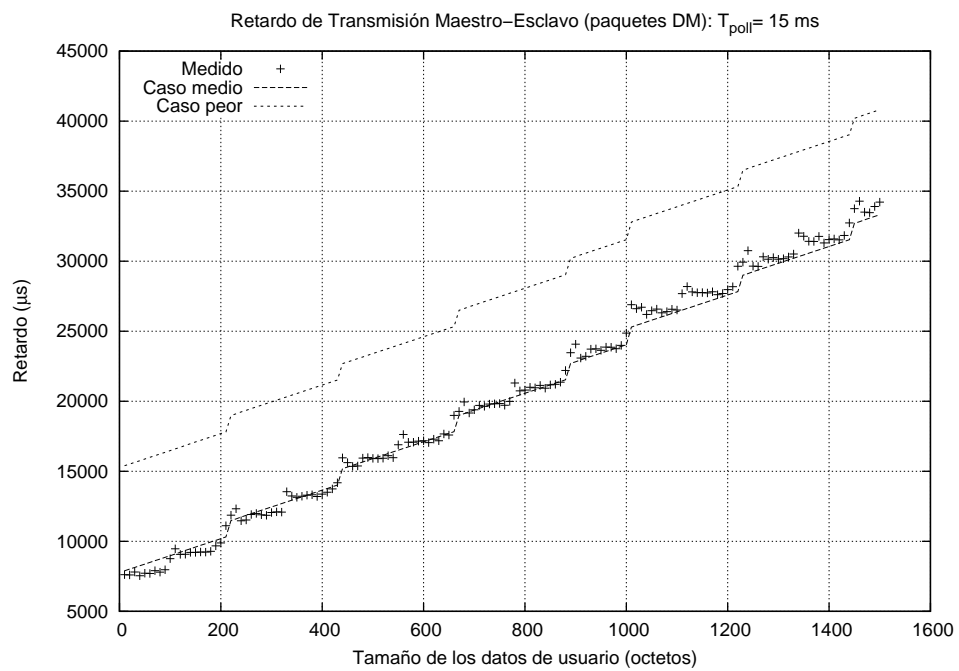


(b) Paquetes DM

Figura 5.15: Comparación de los retardos de transmisión maestro-esclavo medido y teórico corregido para un valor del parámetro T_{poll} de 10 ms



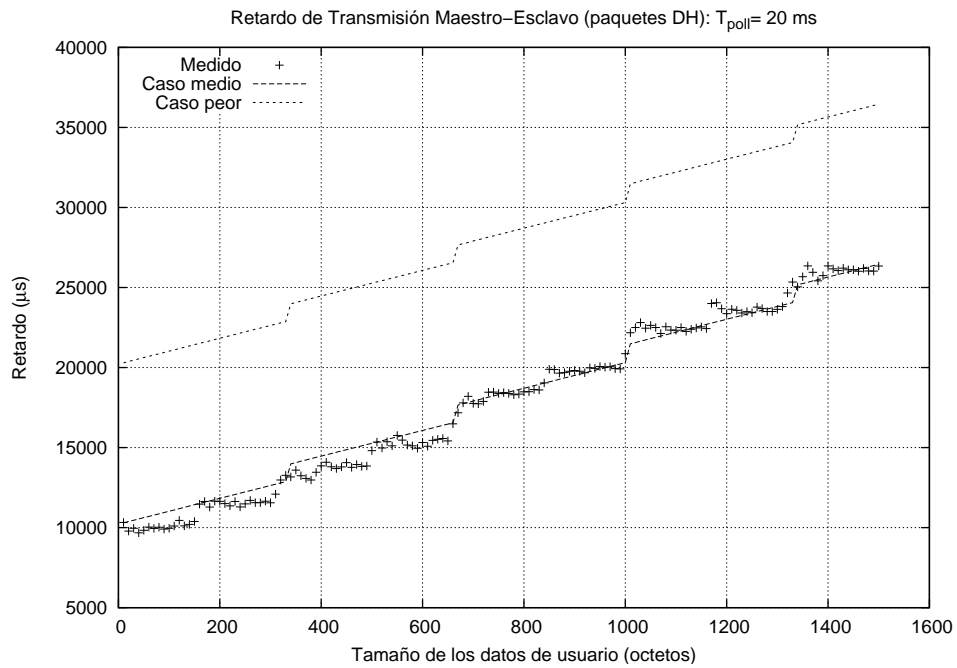
(a) Paquetes DH



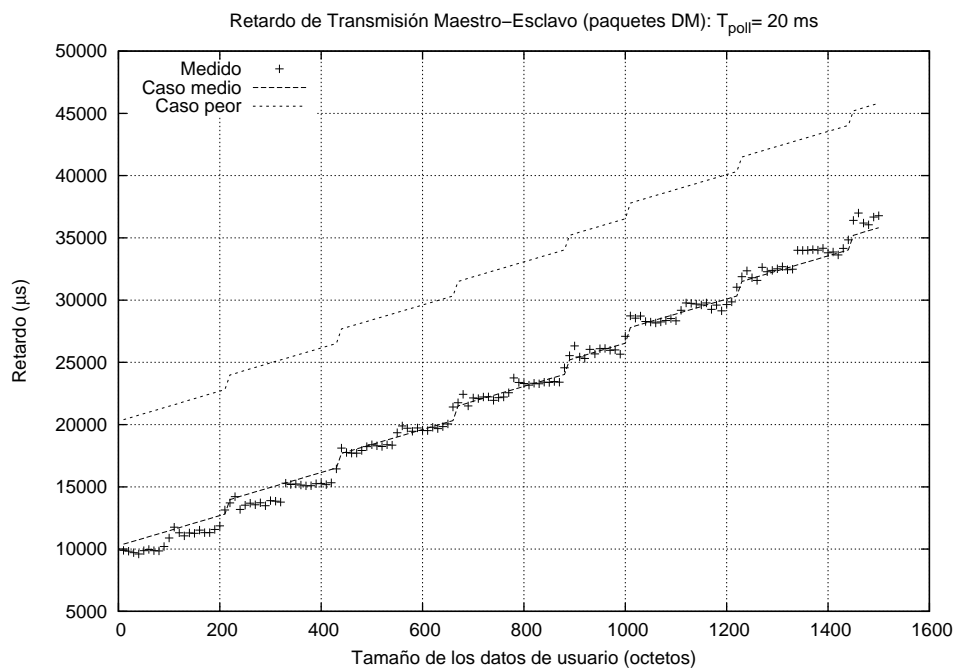
(b) Paquetes DM

Figura 5.16: Comparación de los retardos de transmisión maestro-esclavo medido y teórico corregido para un valor del parámetro T_{poll} de 15 ms

5.3 Efecto de la configuración de la calidad de servicio



(a) Paquetes DH



(b) Paquetes DM

Figura 5.17: Comparación de los retardos de transmisión maestro-esclavo medido y teórico corregido para un valor del parámetro T_{poll} de 20 ms

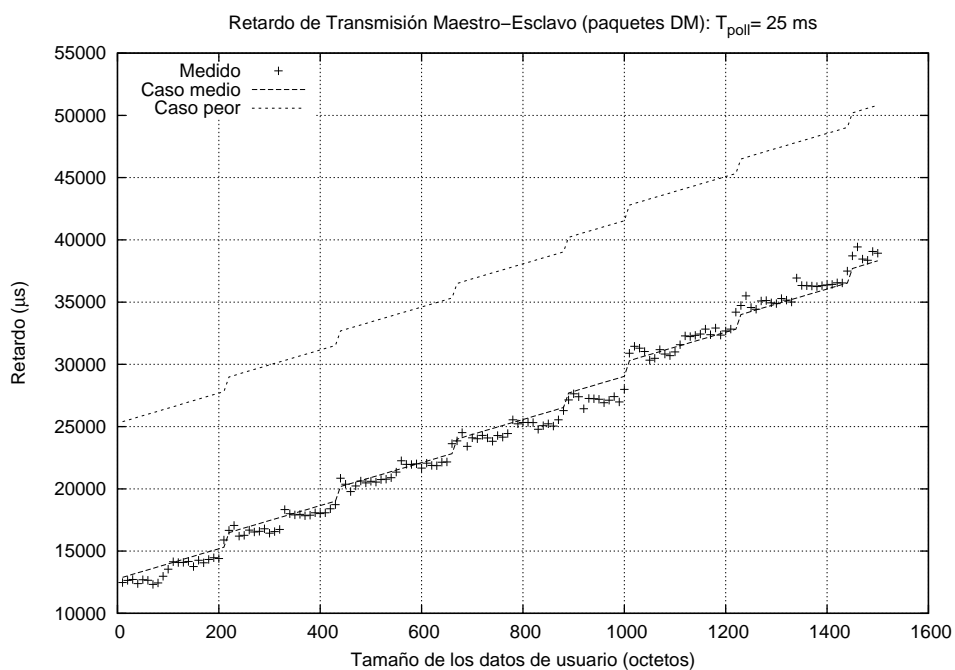
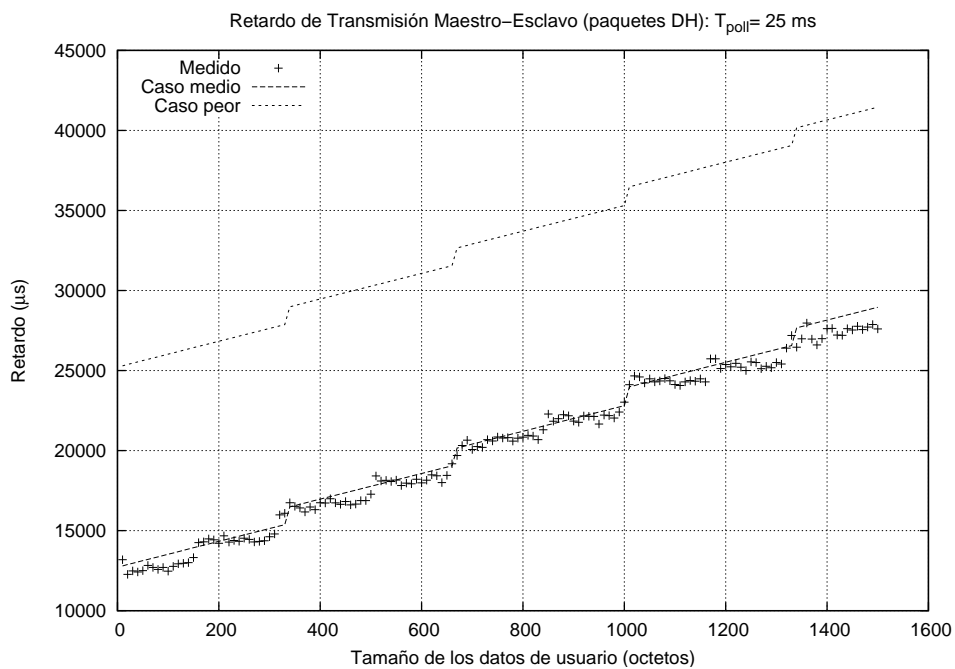


Figura 5.18: Comparación de los retardos de transmisión maestro-esclavo medido y teórico corregido para un valor del parámetro T_{poll} de 25 ms

Para reflejar este comportamiento, es posible afinar el modelo modificando la ecuación definida para el retardo de transmisión $t_{TX}(N)$, tal y cómo se indica a continuación:

$$t_{TX}(N) = \begin{cases} 0 & N = 0 \\ T_S & 0 < N \leq L_1 \\ 3 \cdot T_S & L_1 < N \leq L_3 \\ 5 \cdot T_S & L_3 < N \leq L_5 \\ t_{ACK}(L_5) \cdot \left\lfloor \frac{N}{L_5} \right\rfloor + & N > L_5 \\ +t_{TX}(N \bmod L_5) & \end{cases} \quad (5.3)$$

La modificación propuesta para $t_{TX}(N)$ refleja el hecho de que, cuando el maestro transmite, $t_{TX}(N)$ no depende exactamente del número de bits a transmitir sino del número de *slots* ocupados por el paquete que se utiliza.

En definitiva, este ajuste surge a raíz de considerar, para el cálculo del retardo extremo a extremo, que el esclavo no entrega los datos al *Host* (PC) hasta que no transcurre, al menos, el tiempo correspondiente a un paquete de 1, 3 ó 5 *slots*, completo.

En las figuras 5.19, 5.20, 5.21, 5.22 y 5.23 puede comprobarse que el ajuste realizado (referido como *corregido* y *mejorado*) se adapta mejor al retardo medido para todos los casos, coincidiendo para un intervalo de *poll* de 5 ms, con la gráfica correspondiente al caso peor, y para los restantes valores de T_{poll} , con la obtenida en el caso medio.

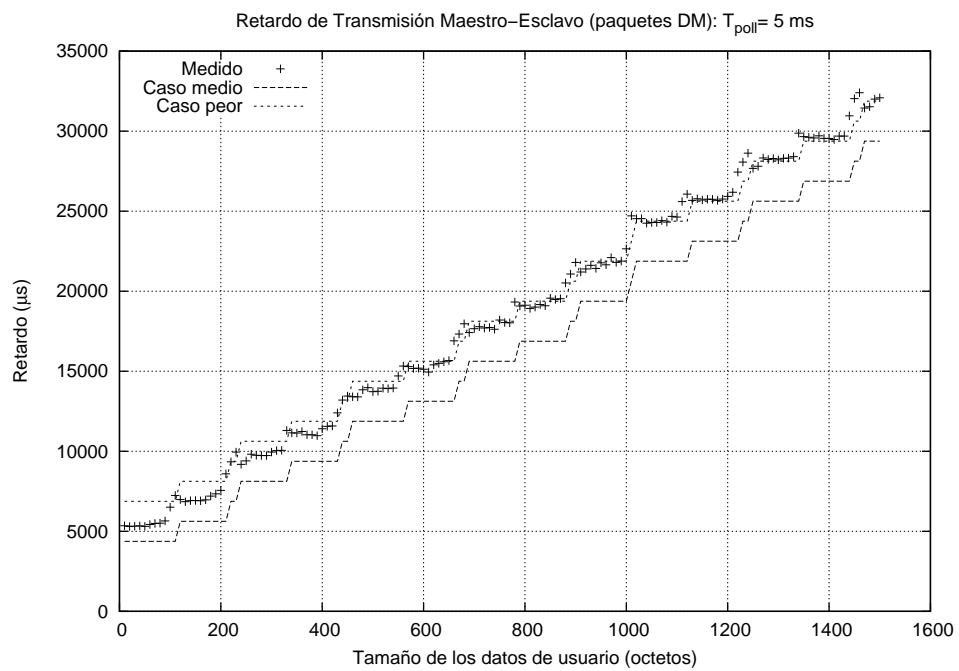
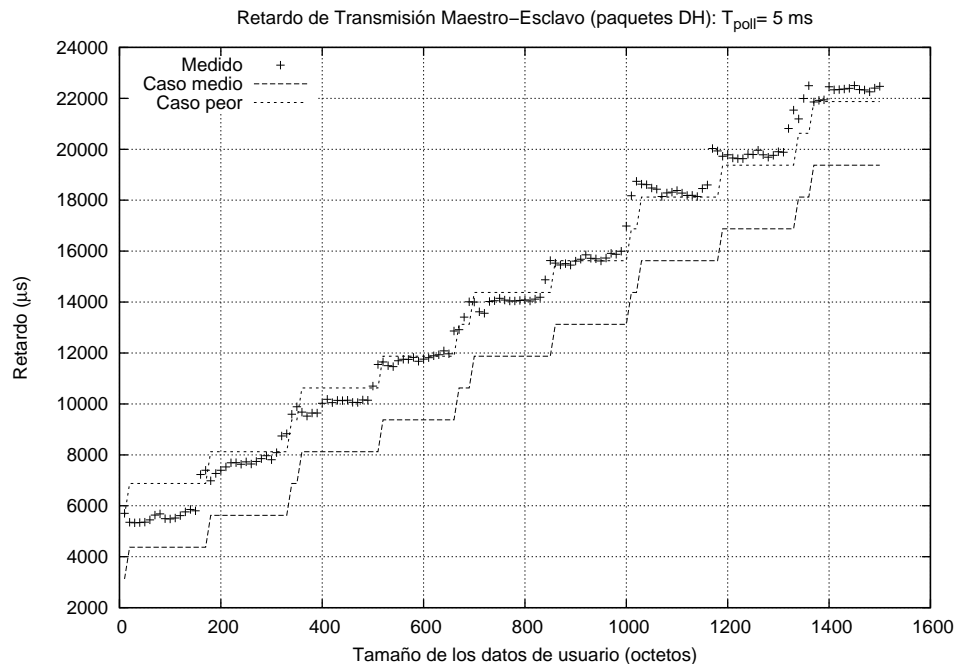
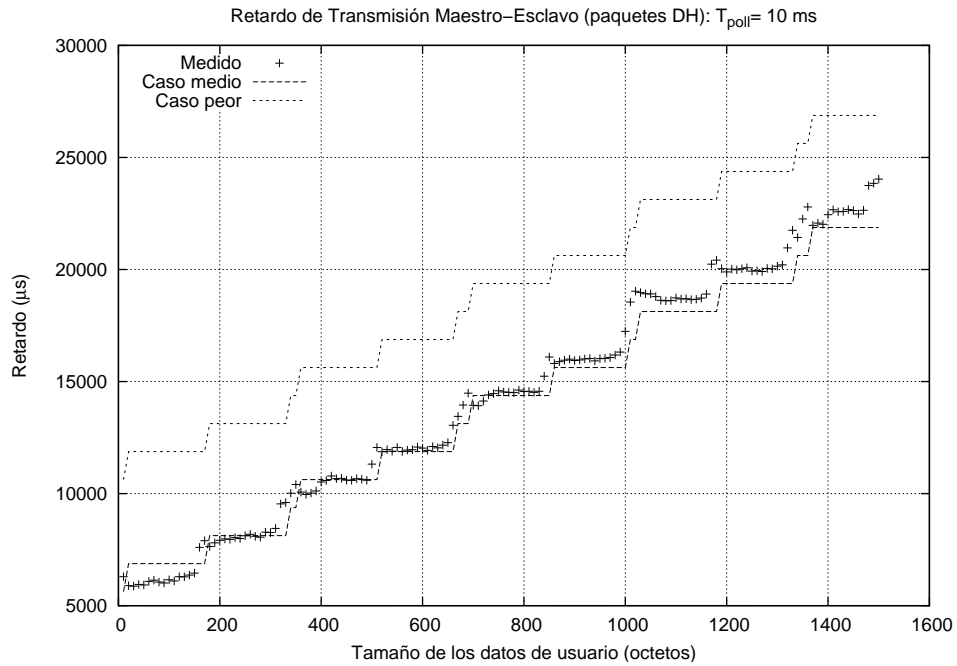
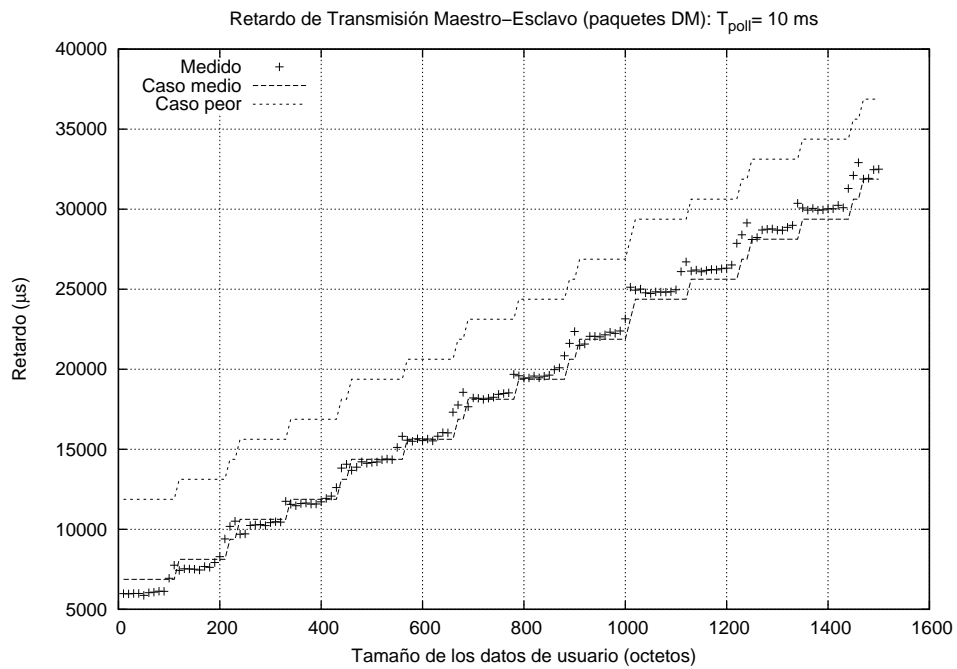


Figura 5.19: Comparación de los retardos de transmisión maestro-esclavo medido y teórico, corregido y mejorado, para un valor del parámetro T_{poll} de 5 ms

5.3 Efecto de la configuración de la calidad de servicio



(a) Paquetes DH



(b) Paquetes DM

Figura 5.20: Comparación de los retardos de transmisión maestro-esclavo medido y teórico, corregido y mejorado, para un valor del parámetro T_{poll} de 10 ms

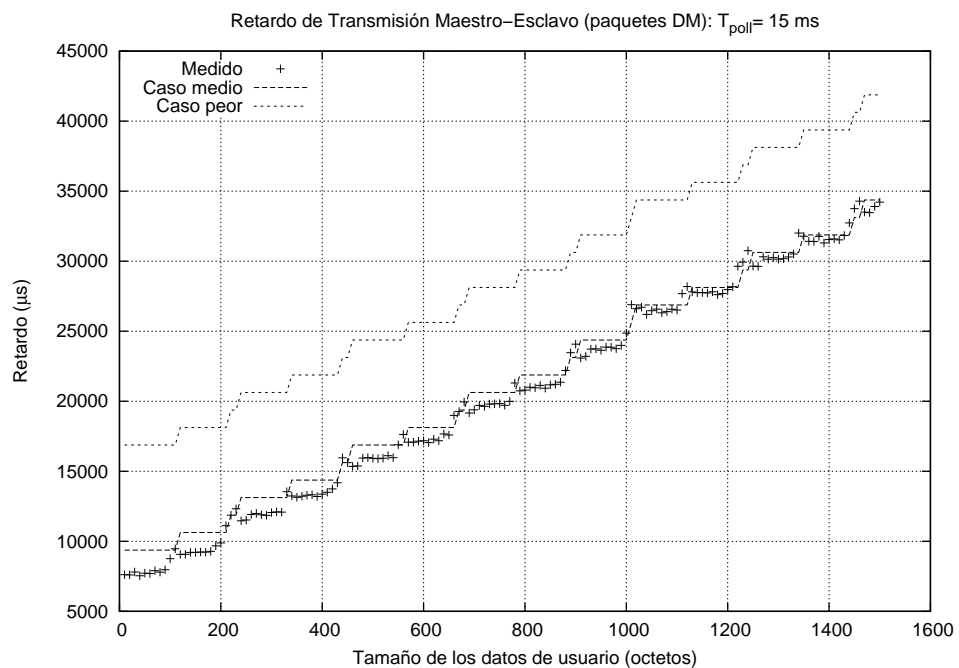
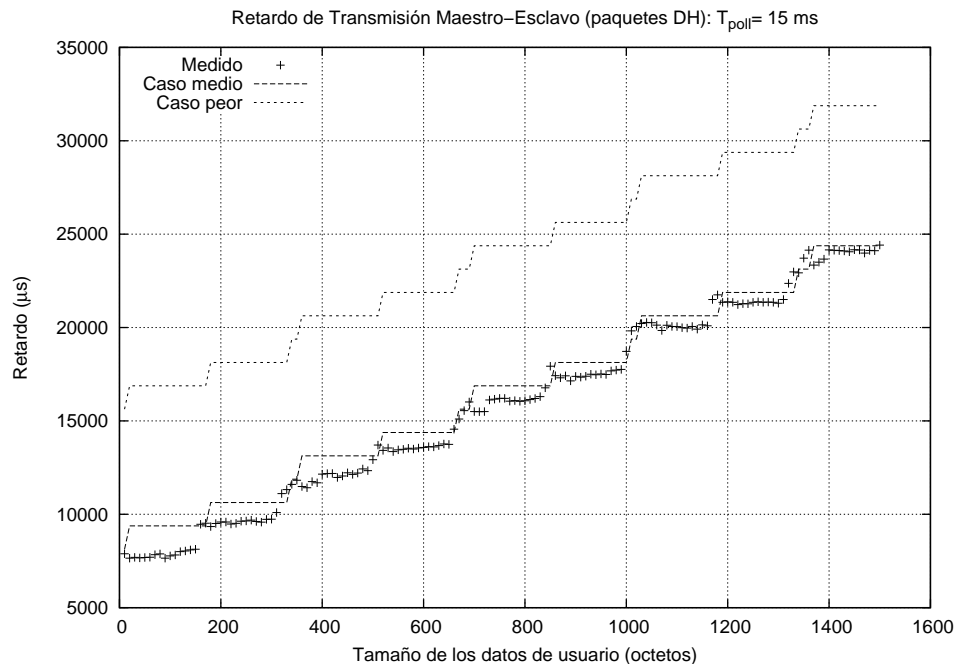
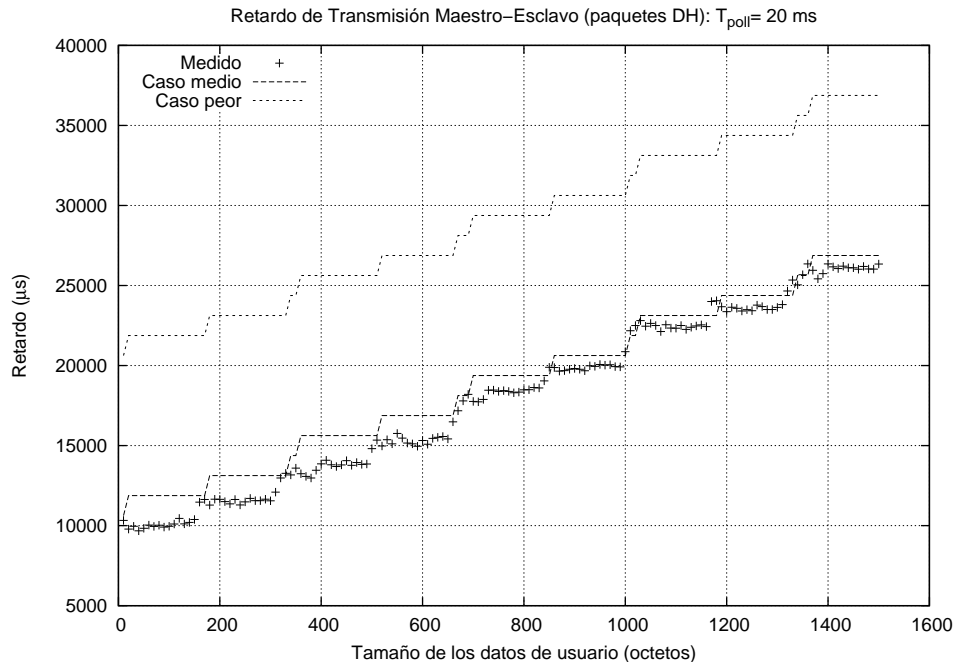
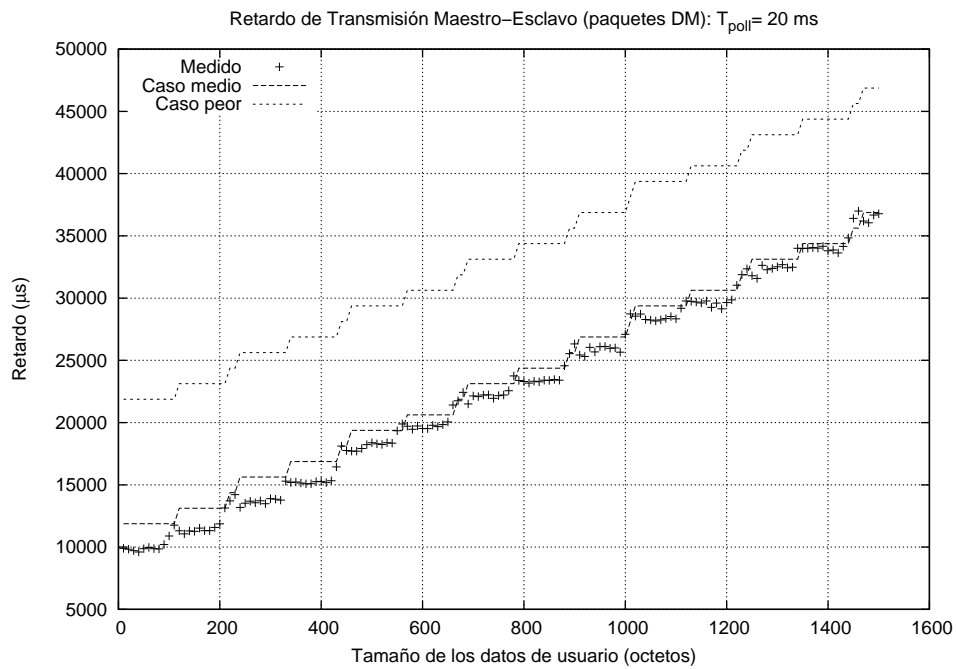


Figura 5.21: Comparación de los retardos de transmisión maestro-esclavo medido y teórico, corregido y mejorado, para un valor del parámetro T_{poll} de 15 ms

5.3 Efecto de la configuración de la calidad de servicio



(a) Paquetes DH



(b) Paquetes DM

Figura 5.22: Comparación de los retardos de transmisión maestro-esclavo medido y teórico, corregido y mejorado, para un valor del parámetro T_{poll} de 20 ms

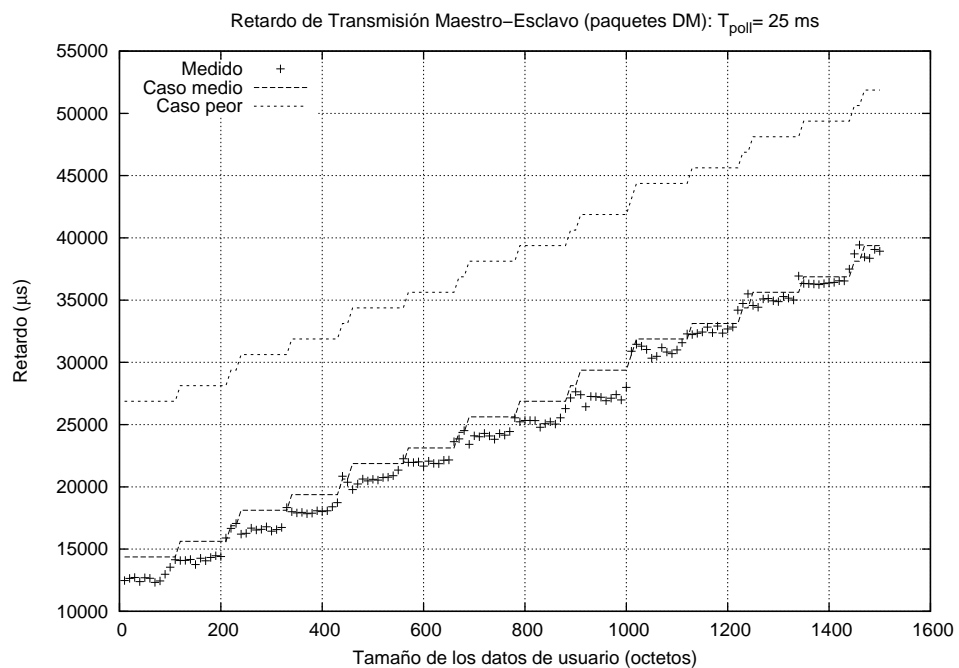
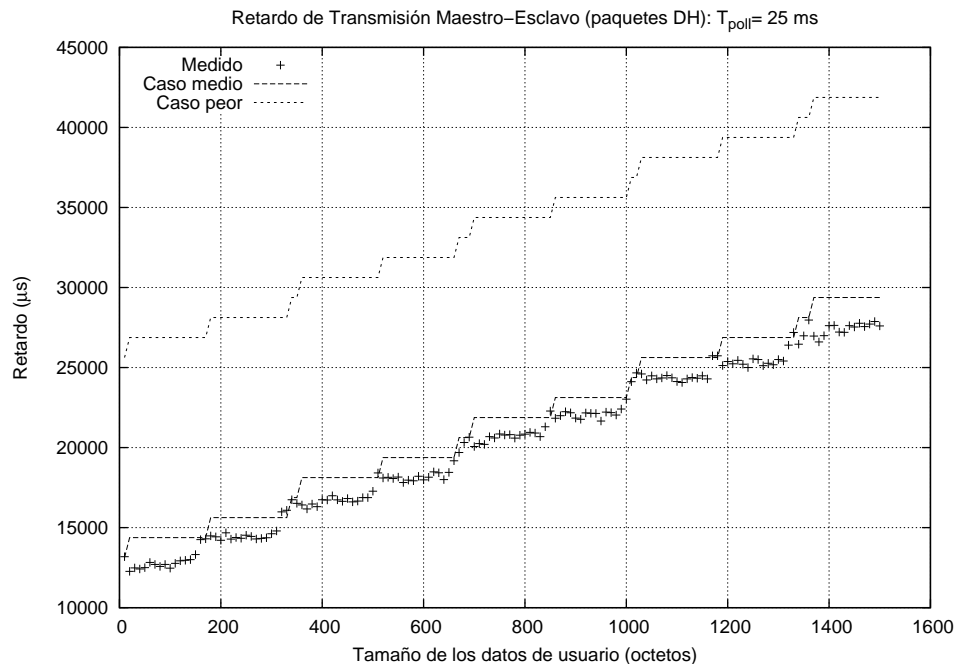


Figura 5.23: Comparación de los retardos de transmisión maestro-esclavo medido y teórico, corregido y mejorado, para un valor del parámetro T_{poll} de 25 ms

No obstante, en las figuras 5.21–5.23, a partir de un intervalo de *poll* de 15 ms se aprecia un error de sobrestimación del retardo medio en torno a 1 ms. Restando este error a la cota media, se consigue aproximar aún más la tendencia, tal y como se observa en las figuras 5.24, 5.25 y 5.26.

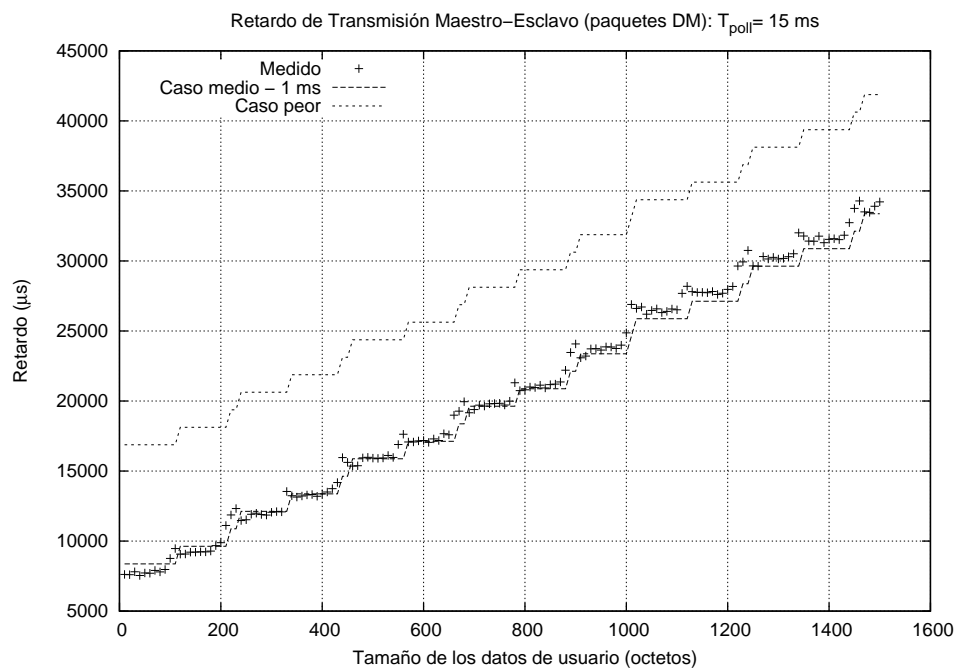
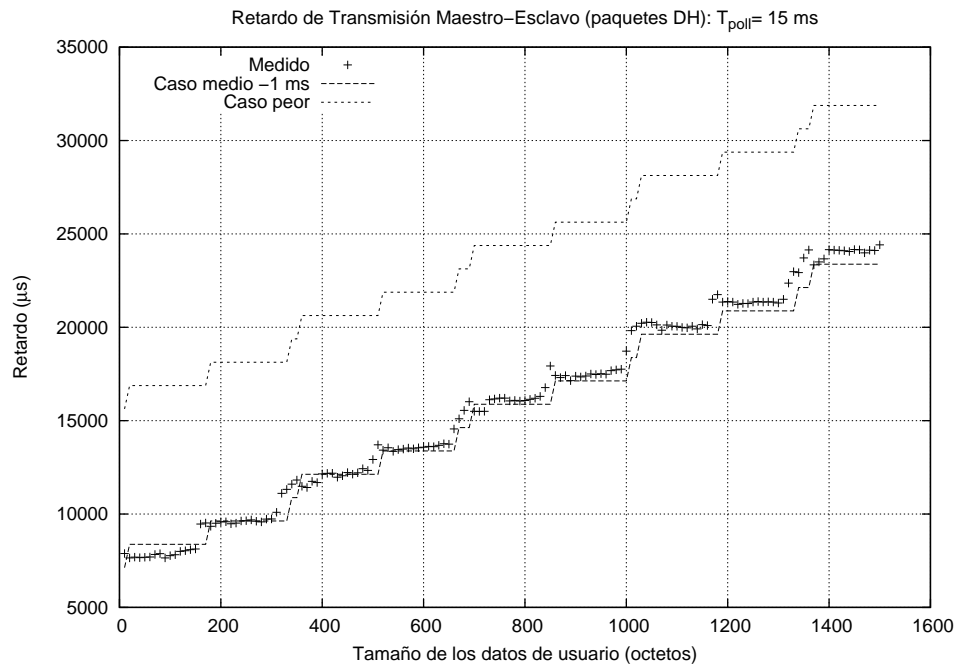
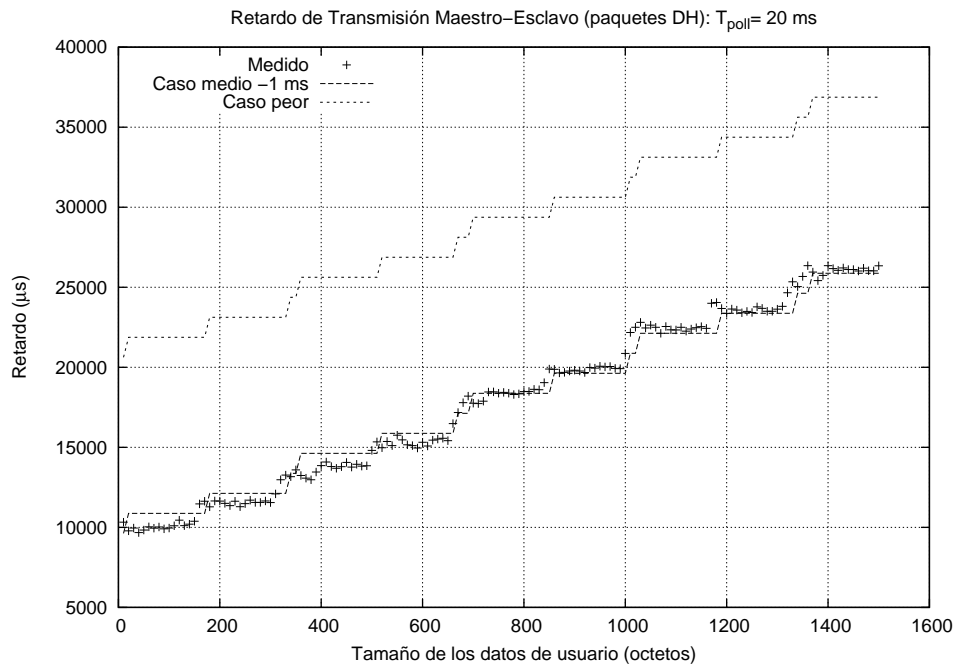
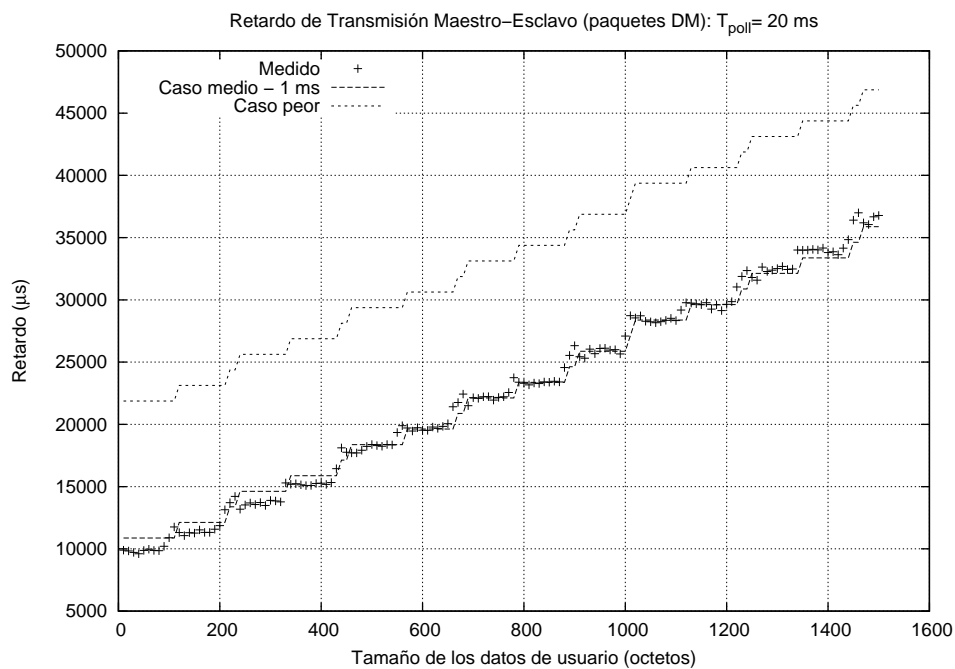


Figura 5.24: Comparación de los retardos de transmisión maestro-esclavo medido y teórico, corregido y mejorado con un desplazamiento de -1 ms , para un valor del parámetro T_{poll} de 15 ms

5.3 Efecto de la configuración de la calidad de servicio

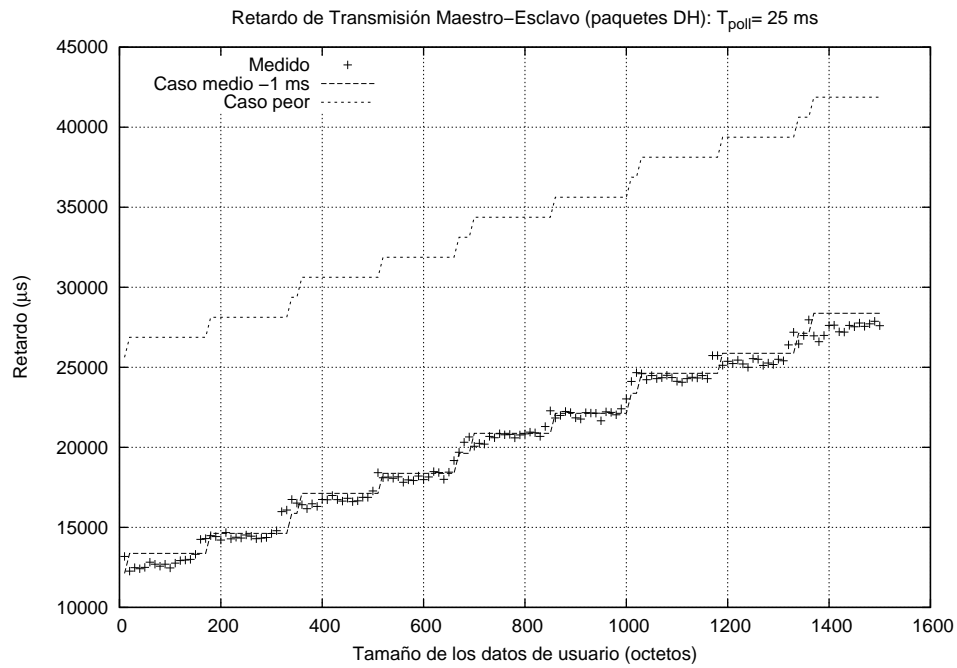


(a) Paquetes DH

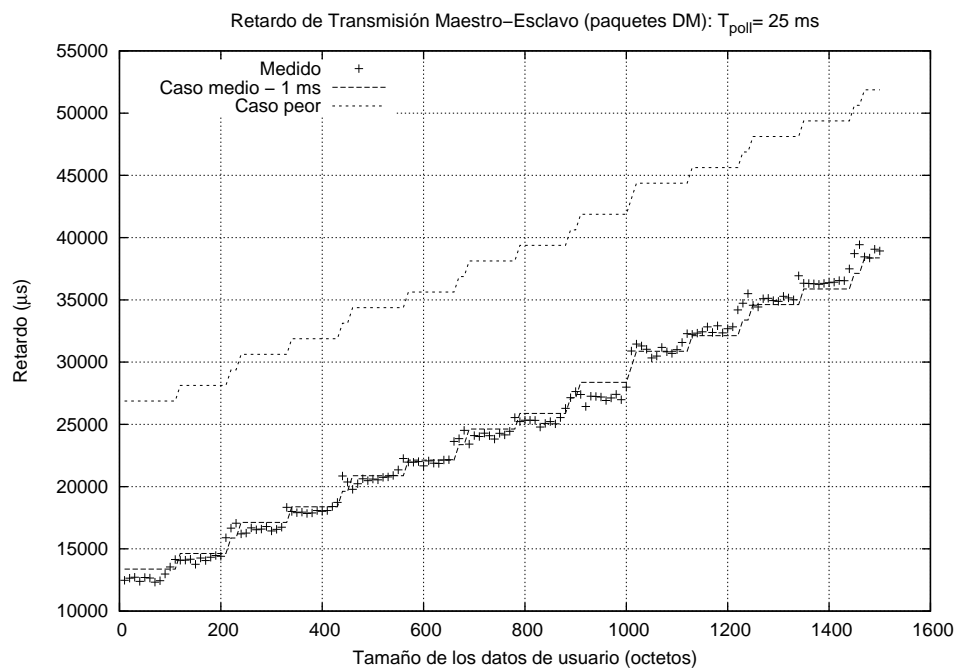


(b) Paquetes DM

Figura 5.25: Comparación de los retardos de transmisión maestro-esclavo medido y teórico, corregido y mejorado con un desplazamiento de -1 ms , para un valor del parámetro T_{poll} de 20 ms



(a) Paquetes DH



(b) Paquetes DM

Figura 5.26: Comparación de los retardos de transmisión maestro-esclavo medido y teórico, corregido y mejorado con un desplazamiento de -1 ms, para un valor del parámetro T_{poll} de 25 ms

5.3.1.5. Corrección del retardo en la transmisión esclavo-maestro

A continuación se presenta la modificación de las ecuaciones que se propusieron en la sección 4.2 para tener en cuenta el efecto de la configuración de calidad en el retardo teórico con el fin de reflejar:

- La influencia real del valor de T_{poll} en el retardo de transmisión descrita en el apartado 5.3.1.2.

Del mismo modo que para el sentido de transmisión maestro-esclavo, la corrección propuesta consiste en considerar que la componente del retardo $t_{ACK}(N)$ no debe diferir de la definida para el maestro, en condiciones ideales, por la ecuación 4.1, ya que de acuerdo con el resultado de las medidas, el tiempo de confirmación de los paquetes Bluetooth que transportan segmentos intermedios de una PDU L2CAP no depende del valor de T_{poll} negociado.

- La diferencia existente en el salto incremental que se produce tras el llenado del primer paquete de 5 *slots* (tal y como se acaba de ver en el apartado 5.3.1.3).

El ajuste propuesto para contemplar este efecto consiste en incluir un *offset* que se añadirá sólo en caso de que para la transmisión de los datos se requiera más de un paquete de cinco *slots*, tal y como se ilustra en la figura 5.27. 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. Para este *offset*, teniendo en cuenta que, como se dijo en el apartado 5.3.1.2, era superior a $2 \cdot T_{SLOT}$, se ha propuesto, finalmente, un valor de $4 \cdot T_{SLOT}$ para modelar que la confirmación del primer paquete de

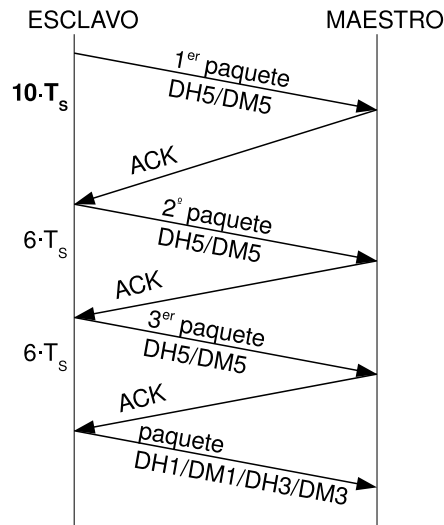


Figura 5.27: Ejemplo que muestra la corrección introducida en el modelo para contemplar, en el sentido de transmisión esclavo-maestro, el salto incremental que se produce tras el llenado del primer paquete de 5 slots

cinco slots se produce no en el slot inmediatamente posterior a la recepción, sino incluso, en algunos casos, hasta cuatro slots más tarde.

Es conveniente resaltar que este *offset* se ha incluido en la ecuación de $t_{R_E}(N)$, correspondiente al retardo global calculado para el nivel RFCOMM (perfil SPP), y no en la componente $t_{ACK}(N)$ porque no se suma por cada paquete DH5 que se transmita.

Así, para el caso peor, resulta la siguiente ecuación 5.4:

$$t_{R_E}(N) = T_S + \delta(N) + t_{R_M}(N) \quad (5.4)$$

y para el caso medio:

$$\overline{t_{R_E}}(N) = T_S + \delta(N) + \overline{t_{R_M}}(N) \quad (5.5)$$

donde $\delta(N)$, 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:

$$\delta(N) = \begin{cases} 4 \cdot T_S & \text{si } n_{fnf_R}(N) > 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5.6)$$

Las ecuaciones 5.4 y 5.5 se diferencian de las equivalentes definidas para el maestro (5.1 y 5.2), además de la inclusión del *offset*, en el *slot* de direccionamiento que el esclavo tiene que esperar para comenzar a transmitir.

En las figuras 5.28, 5.29, 5.30, 5.31 y 5.32 se compara el retardo de transmisión obtenido teóricamente mediante las ecuaciones 5.5 y 5.4 para los valores del parámetro *Latency* indicados en la tabla 5.3, a partir de 5000 μs , con los medidos en el sentido esclavo-maestro, presentados ya en el apartado 5.3.1.2 (véase la figura 5.8).

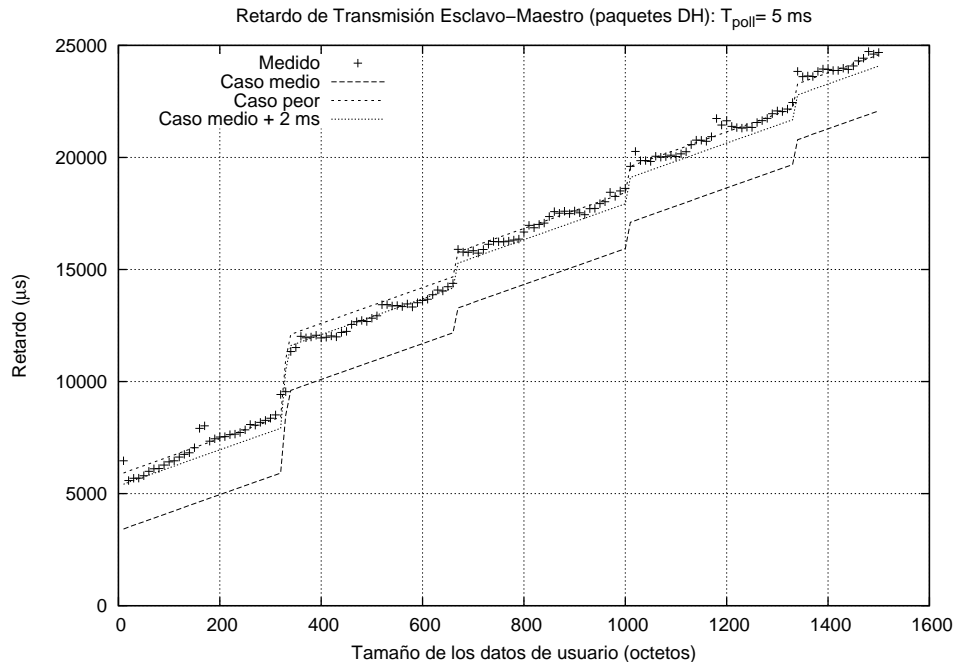
En las figuras 5.28–5.32 también se ha representado la curva resultante de sumar al retardo medio (ecuación 5.5) los 2 ms, en los que, como se indicó en el apartado 5.3.1.2, se supera el desplazamiento de $T_{poll}/2$, que sí es suficiente para la transmisión maestro-esclavo. Así, puede comprobarse que la corrección realizada se ajusta al retardo medido para todos los casos, con la salvedad de que existe una pequeña deriva (inferior a 1 ms), que, en el caso de que la transmisión se realice con paquetes DM, tiene lugar para tamaños de datos superiores a 890 octetos y que, para paquetes DH sucede a partir de 1390 octetos.

En el caso de los paquetes DH, la razón que puede justificarlo es que 1390 octetos exceden la capacidad del *buffer* del módulo *hardware*, que en el caso del

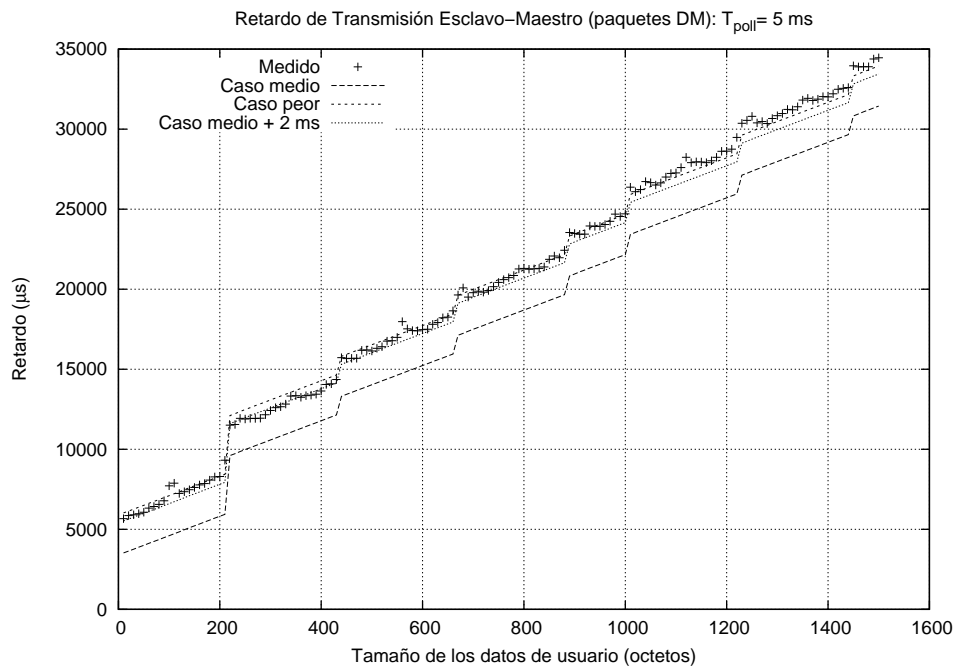
circuito integrado de CSR es de 8 posiciones de 192 octetos. El motivo es el siguiente: 1390 octetos se envían como dos tramas RFCOMM, de 1008 octetos y 382 octetos, que con el *overhead* añadido por RFCOMM y L2CAP, se convierten, respectivamente, en dos PDUs de L2CAP de 1017 octetos y 391 octetos. Para su envío al módulo *hardware* a través del HCTL, los 1017 octetos se transfieren como 5 fragmentos de 192 octetos (teniendo en cuenta el tamaño de cada una de las posiciones del *buffer*) y 1 de 57 octetos, y los 391 octetos, como 2 fragmentos de 192 octetos y 1 de 7 octetos. Pues bien, este último fragmento quedará retenido en el *Host* hasta que se libere alguna posición del *buffer*.

En cuanto al hecho de que la divergencia, al utilizar paquetes DM, se produzca antes del límite de los 1390 octetos, cabe argumentar que podría atribuirse a que el procesamiento que el nivel BB lleva a cabo para proteger los datos mediante codificación FEC $2/3$, consume más tiempo conforme mayor sea el número de octetos a codificar.

5.3 Efecto de la configuración de la calidad de servicio



(a) Paquetes DH



(b) Paquetes DM

Figura 5.28: Comparación de los retardos de transmisión esclavo-maestro medidos y teórico corregido para un valor del parámetro T_{poll} de 5 ms

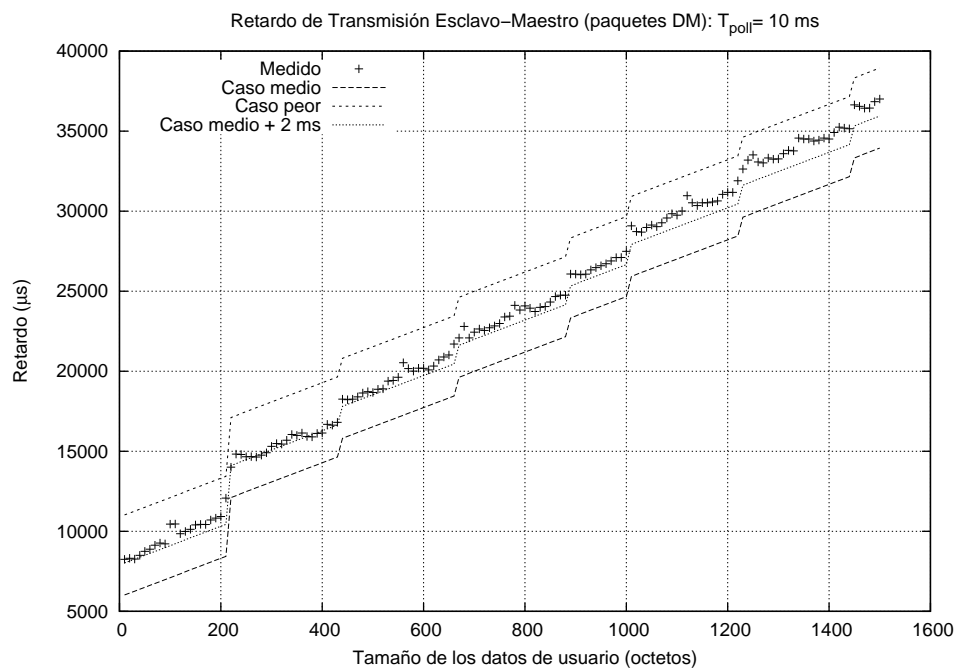
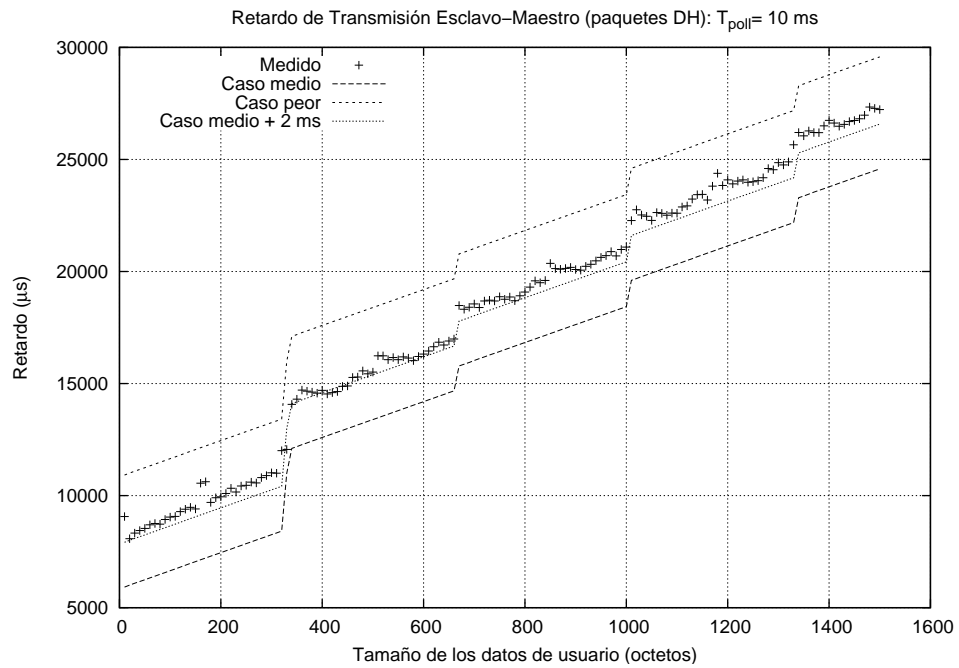
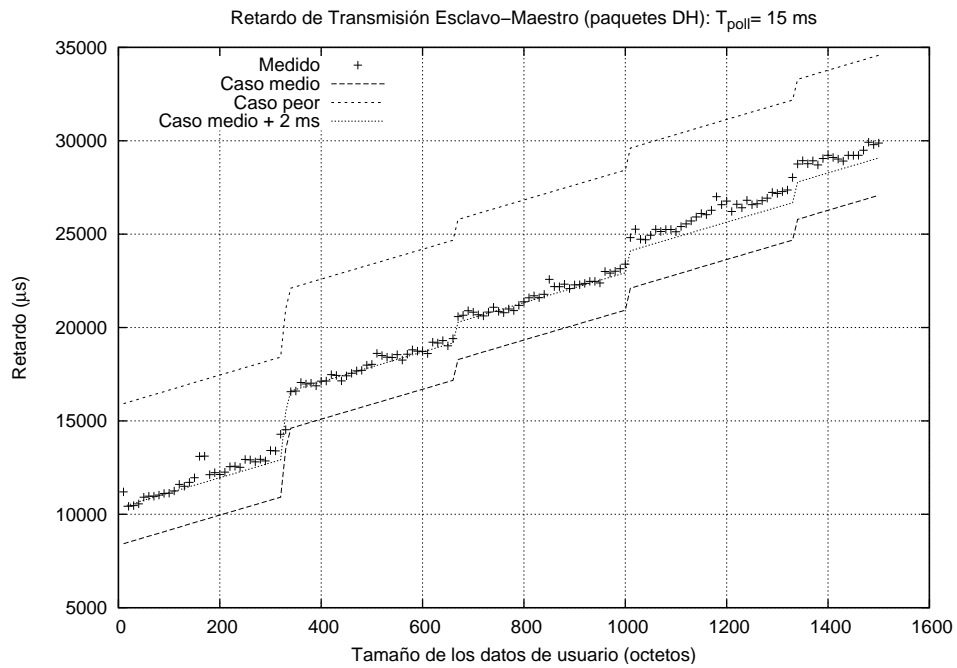
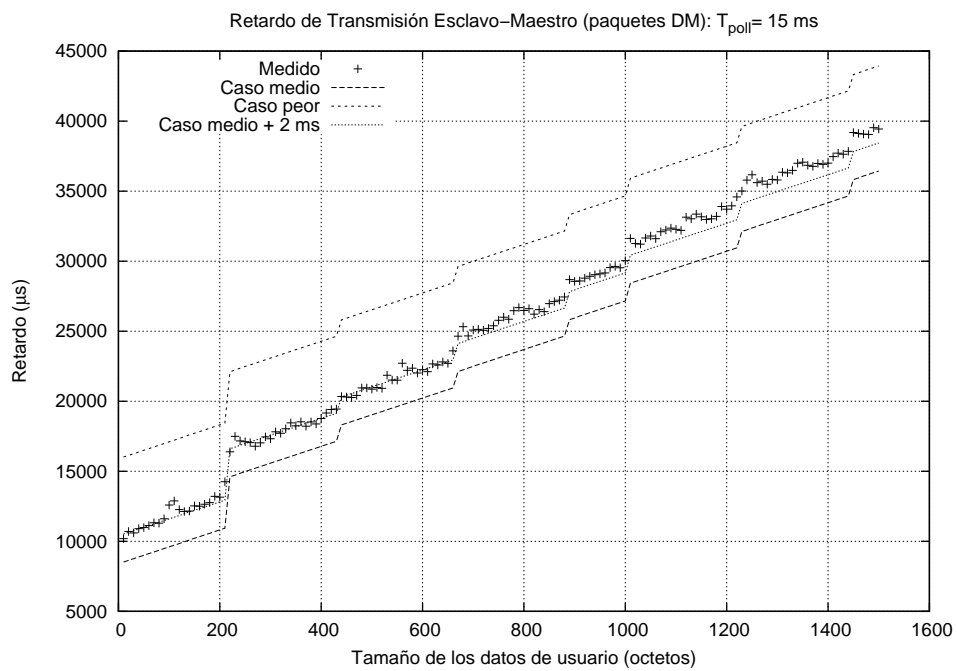


Figura 5.29: Comparación de los retardos de transmisión esclavo-maestro medidos y teórico corregido para un valor del parámetro T_{poll} de 10 ms

5.3 Efecto de la configuración de la calidad de servicio



(a) Paquetes DH



(b) Paquetes DM

Figura 5.30: Comparación de los retardos de transmisión esclavo-maestro medidos y teórico corregido para un valor del parámetro T_{poll} de 15 ms

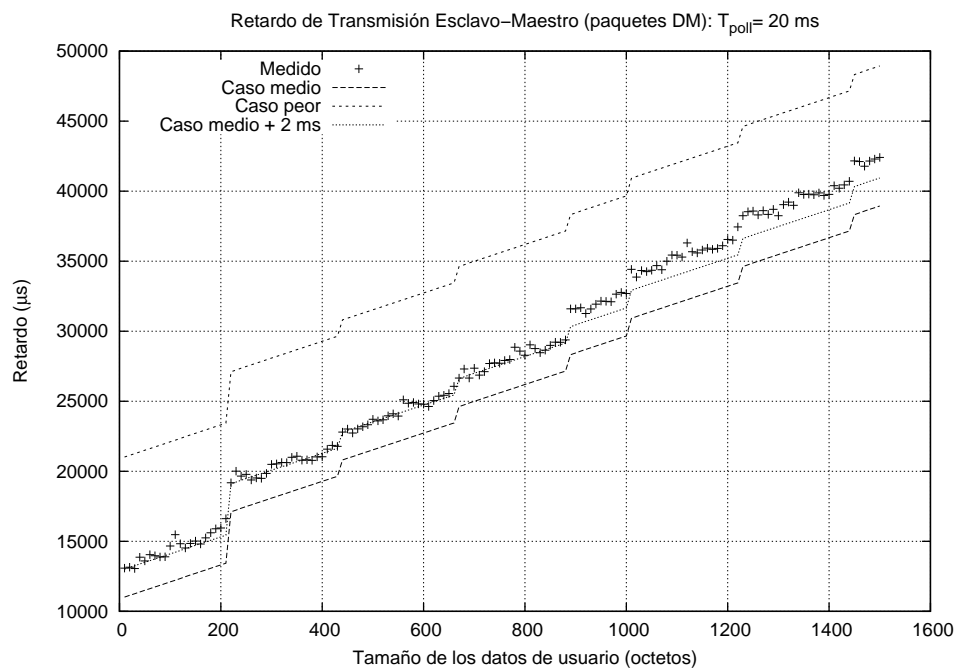
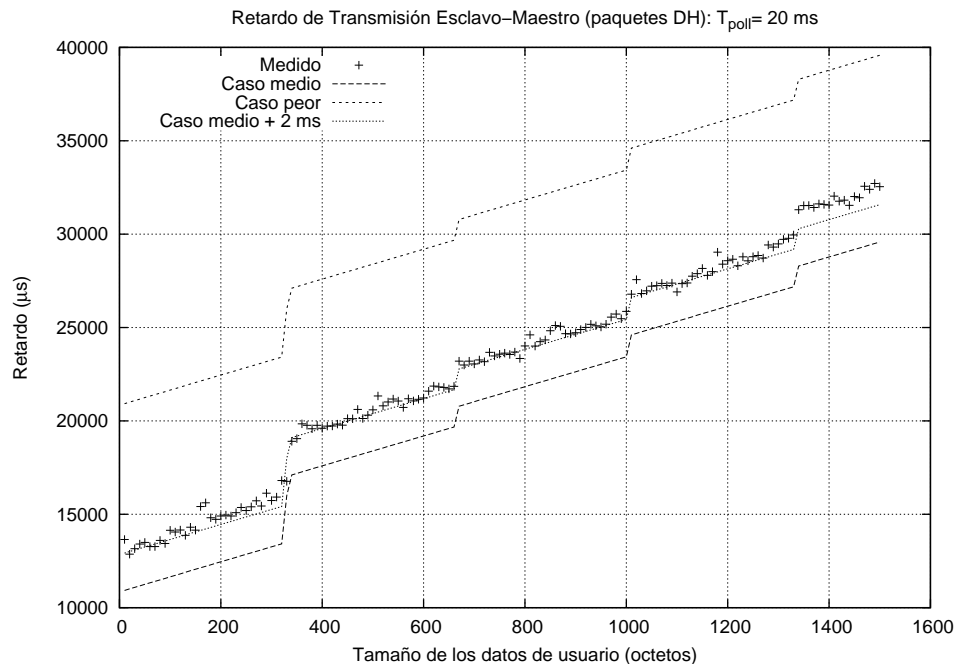
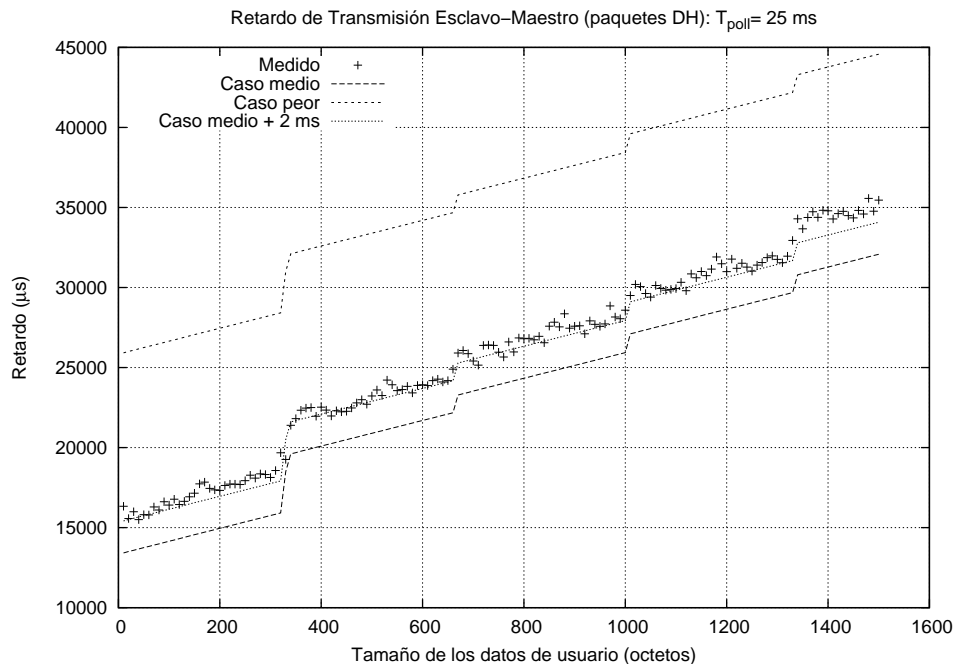
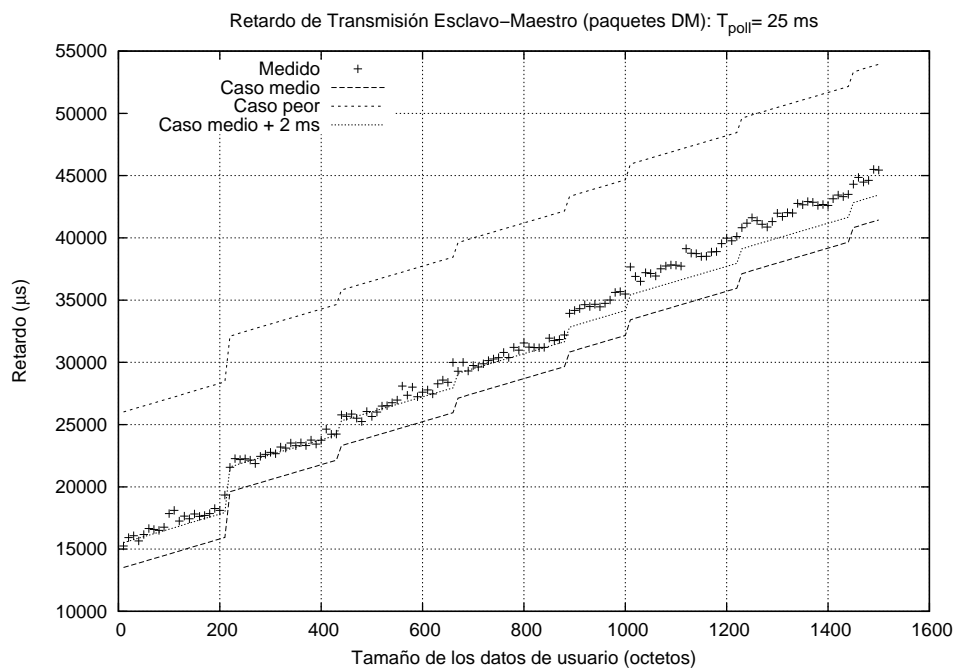


Figura 5.31: Comparación de los retardos de transmisión esclavo-maestro medidos y teórico corregido para un valor del parámetro T_{poll} de 20 ms

5.3 Efecto de la configuración de la calidad de servicio



(a) Paquetes DH



(b) Paquetes DM

Figura 5.32: Comparación de los retardos de transmisión esclavo-maestro medidos y teórico corregido con paquetes DM para un valor del parámetro T_{poll} de 25 ms

5.3.2. Medidas para una *piconet* de tres dispositivos

En el apartado anterior se ha evaluado el efecto de la configuración de calidad para una *piconet* de dos dispositivos, demostrando que, en presencia de un solo esclavo, las transmisiones de datos que requieren más de un paquete de 5 *slots* no se retardan un tiempo de T_{poll} por cada paquete que transporte un fragmento intermedio. Sin embargo, al aumentar el número de dispositivos participantes en la *piconet*, teniendo en cuenta que el servicio proporcionado es de tipo *Best Effort* y que el maestro atiende a los esclavos siguiendo una planificación *Round Robin* (según especifica el fabricante CSR [CSR]), se plantea el reto de, al menos, acotar el retardo de transmisión teniendo en cuenta el incremento del número de esclavos y el valor de T_{poll} configurado.

La consecución de este objetivo se ha llevado a cabo de forma gradual: en primer lugar, en este apartado se perfila el modelo de comportamiento para el caso particular de una *piconet* de dos esclavos. En una segunda fase, se extrapola dicho modelo a una *piconet* de varios esclavos (véase el apartado 5.3.3). Finalmente, en el apartado 5.3.4 se verifica la validez del modelo para una *piconet* de tres esclavos.

Seguidamente, se comienza proponiendo cotas mínima y máxima del retardo de transmisión para el perfil SPP y una *piconet* de dos esclavos (apartado 5.3.2.1), para presentar después los resultados de las medidas que permiten comprobar que el retardo medido en cada sentido de la transmisión queda comprendido entre estas cotas (apartado 5.3.2.2).

5.3.2.1. Modelo teórico para una *piconet* de dos esclavos

La componente del retardo que se va a ver claramente afectada al existir más de un esclavo es $t_{ACK}(N)$, que ha de redefinirse teniendo presente que en las especificaciones del fabricante CSR [CSR] se indica que el algoritmo de planificación es RR. Por consiguiente, al haber un segundo esclavo, el maestro va a direccionar a los dos esclavos, alternadamente, pudiendo darse las dos situaciones que a continuación se describen:

- Las transmisiones a los esclavos no se sincronizan en el tiempo (es decir, en un momento dado, sólo envía datos un esclavo). El tiempo mínimo que el maestro tardaría en direccionar a un mismo esclavo sería: el número de *slots* ocupados por la transmisión al esclavo E1 (1, 3 ó 5 *slots*), 1 *slot* en el que el esclavo E1 confirma el paquete recibido, 1 *slot* de direccionamiento al esclavo E2 con un paquete POLL por la ausencia de datos pendientes de transmisión y 1 *slot* en el que el esclavo E2 responde al maestro. Así, la cota mínima de la componente $t_{ACK}(N)$, $t_{ACK_{min}}(N)$, tendría que ser redefinida de acuerdo con la siguiente ecuación:

$$t_{ACK_{min}}(N) = \begin{cases} 0 & N = 0 \\ 4 \cdot T_s & 0 < N \leq L_1 \\ 6 \cdot T_s & L_1 < N \leq L_3 \\ 8 \cdot T_s & L_3 < N \leq L_5 \\ (8 \cdot T_s) \cdot \left\lfloor \frac{N}{L_5} \right\rfloor + \\ + t_{ACK_{min}}(N \bmod L_5) & N > L_5 \end{cases} \quad (5.7)$$

La figura 5.33 ilustra cómo se vería afectada la transmisión de datos de longitud superior a la capacidad de un paquete de 5 *slots* en el caso descrito.

- Las peticiones de datos se sincronizan en el tiempo. Las transmisiones de datos a los esclavos se sucederán en secuencia, de forma que, en el peor de los casos, tras la recepción de la confirmación de los datos transmitidos al esclavo E1 en un paquete de 1, 3 ó 5 *slots*, el maestro procederá a enviar al esclavo E2 un paquete de 5 *slots*. Por consiguiente, la cota máxima de la componente $t_{ACK}(N)$, $t_{ACK_{max}}(N)$, vendrá dada por la siguiente ecuación:

$$t_{ACK_{max}}(N) = \begin{cases} 0 & N = 0 \\ 8 \cdot T_s & 0 < N \leq L_1 \\ 10 \cdot T_s & L_1 < N \leq L_3 \\ 12 \cdot T_s & L_3 < N \leq L_5 \\ (12 \cdot T_s) \cdot \left\lfloor \frac{N}{L_5} \right\rfloor + \\ + t_{ACK_{max}}(N \bmod L_5) & N > L_5 \end{cases} \quad (5.8)$$

En este caso, la separación entre transmisiones de fragmentos consecutivos de cada uno de los esclavos (E1 y E2) pasa a ser de 12 *slots*, como se muestra en la figura 5.34. Nótese que en estas condiciones, no se puede garantizar un intervalo de *poll* de 8 *slots*, correspondiente a un valor del parámetro *Latency* de 5000 μs .

Las ecuaciones $t_{ACK_{min}}(N)$ y $t_{ACK_{max}}(N)$ definidas para el sentido de transmisión maestro-esclavo son también válidas para el caso en el que sea el esclavo el que transmite. Es decir, una vez que el esclavo es direccionado y envía el

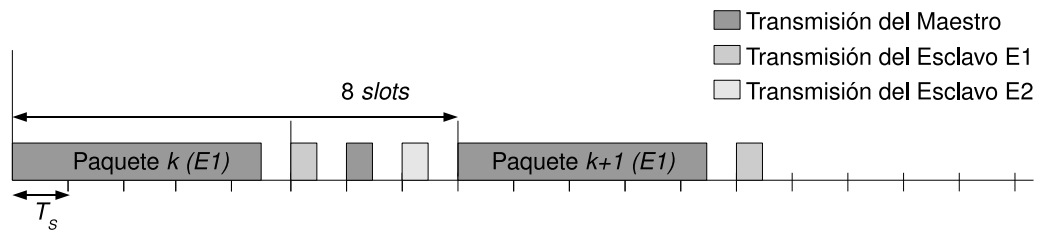


Figura 5.33: Tiempo mínimo entre la transmisión de fragmentos consecutivos a un mismo esclavo en presencia de un segundo esclavo para el que no hay datos pendientes de transmisión

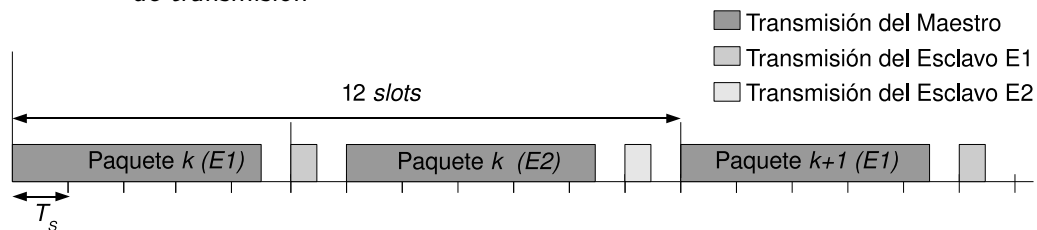


Figura 5.34: Tiempo máximo entre la transmisión de fragmentos consecutivos a un mismo esclavo, debido a la alternancia de las transmisiones a los esclavos E1 y E2 de fragmentos intermedios en paquetes de 5 slots

primer paquete de la secuencia que transportará todos los fragmentos asociados a una misma petición de datos sobre RFCOMM, el espaciado entre paquetes consecutivos que contengan fragmentos intermedios es de 8 slots, si los dos esclavos no tienen datos pendientes de envío cuando son direccionados (véase la figura 5.35), o bien 12 slots, en caso contrario (véase la figura 5.36).

Por último, se procede a obtener las cotas del retardo de transmisión para el perfil SPP en una *piconet* de dos esclavos, para cada uno de los sentidos de comunicación:

Sentido de transmisión maestro-esclavo: La cota mínima del retardo de transmisión para el perfil SPP en una *piconet* de dos esclavos, $\overline{t_{R_{M_{min}}}}(N)$, puede obtenerse a partir del retardo medio definido por la ecuación 5.2, en el caso

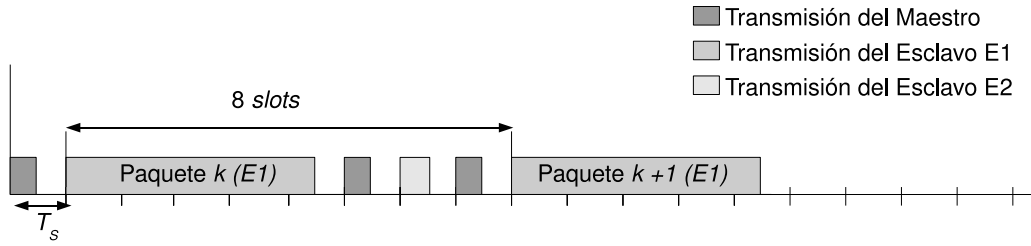


Figura 5.35: Tiempo mínimo entre la transmisión de fragmentos consecutivos de un mismo esclavo en presencia de un segundo esclavo que tiene datos pendientes de transmisión

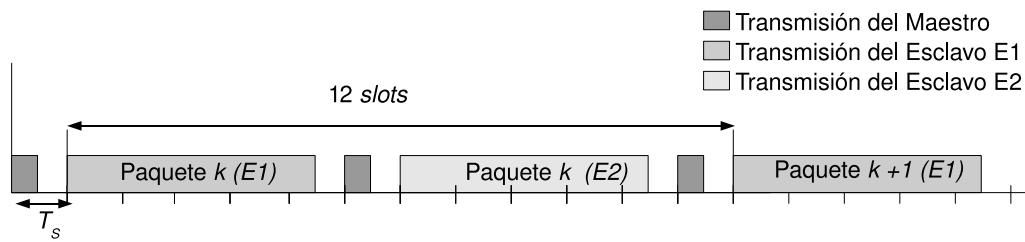


Figura 5.36: Tiempo máximo entre las transmisiones de fragmentos consecutivos de un mismo esclavo, debido a la alternancia de las transmisiones de los esclavos E1 y E2 de fragmentos intermedios en paquetes de 5 slots

de una *piconet* de un único esclavo, reemplazando $t_{ACK}(N)$ por $t_{ACK_{min}}(N)$, de acuerdo con la siguiente ecuación:

$$\begin{aligned} \overline{t_{R_{M_{min}}}}(N) = & (T_{poll}/2) + n_{fnfr}(N) \cdot t_{ACK_{min}}(L_R + O_R(L_R) + C_L) + \\ & + t_{TX}(L_{ffr} + O_R(L_{ffr}) + C_L) \end{aligned} \quad (5.9)$$

Nótese que el *offset* sumado en el primer término de la ecuación anterior ($T_{poll}/2$), al igual que en la *piconet* de un esclavo, 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. Es decir, $T_{poll}/2$ es precisamente la media de una variable aleatoria que sigue una distribución uniforme entre 0 y T_{poll} . Al respecto, procede aclarar que, pese a que se está

introduciendo un *offset* medio, se ha utilizado la designación de cota mínima con el objeto de resaltar que se está considerando la mínima distancia entre fragmentos consecutivos cuando hay dos esclavos en la *piconet*.

Por contra, la definición de la cota superior, $t_{R_{M_{max}}}(N)$, requiere, además de sustituir $t_{ACK}(N)$ por $t_{ACK_{max}}(N)$ en la ecuación 5.1, tener en cuenta que el intervalo de *poll* configurado pueda o no garantizarse, por el envío de paquetes consecutivos. Así, el límite $t_{R_{M_{max}}}(N)$ viene dado por la ecuación siguiente:

$$t_{R_{M_{max}}}(N) = \Delta_{poll} + n_{fnf_R}(N) \cdot t_{ACK_{max}}(L_R + O_R(L_R) + C_L) + t_{TX}(L_{ff_R} + O_R(L_{ff_R}) + C_L) \quad (5.10)$$

donde Δ_{poll} responde a la siguiente expresión:

$$\Delta_{poll} = \max(12 \cdot T_S, T_{poll}) \quad (5.11)$$

La fórmula 5.11, para un valor de T_{poll} de 5 ms, fija el incremento del retardo en un valor de 12 *slots* ($12 \cdot T_S$) con el fin de contemplar el hecho de que, al haber dos esclavos que van a ser direccionados secuencialmente, cuando las transmisiones a los dos esclavos hacen uso de paquetes de 5 *slots* no puede satisfacerse el intervalo de *poll*.

En cuanto a la definición de $t_{TX}(N)$, hay que tener en cuenta que la componente $t_{ACK}(N)$ tendrá que ser sustituida por $t_{ACK_{min}}(N)$ o por $t_{ACK_{max}}(N)$, dependiendo de que se esté calculando la cota inferior o superior.

Sentido de transmisión esclavo-maestro: En este caso las cotas mínimas y máxima del retardo de transmisión para el perfil SPP, $\overline{t_{R_{E_{min}}}}(N)$ y $t_{R_{E_{max}}}(N)$, se

obtienen a partir de las definidas para el maestro:

$$\overline{t_{R_{E_{min}}}}(N) = \overline{t_{R_{M_{min}}}}(N) + T_S \quad (5.12)$$

$$t_{R_{E_{max}}}(N) = t_{R_{M_{max}}}(N) + T_S \quad (5.13)$$

5.3.2.2. Validación del modelo teórico para una *piconet* de dos esclavos

El procedimiento de medida ha consistido en establecer dos conexiones RFCOMM sobre un mismo dispositivo Bluetooth con dos esclavos distintos (esclavos E1 y E2). Para ello, se ha requerido ejecutar dos instancias del servidor RFCOMM, así como dos instancias del cliente RFCOMM, todas en la misma máquina.

Al igual que en el caso de una *piconet* de dos dispositivos, a cada uno de los servidores se le especifica como argumentos de descripción del experimento: el modo de funcionamiento (transmisión o recepción), el rango de tamaños a barrer y la periodicidad con la que se quiere transmitir.

En concreto, se han establecido dos conexiones RFCOMM y se ha medido el retardo para un periodo de transmisión variable de acuerdo con una distribución uniforme entre 90 y 100 ms. Para distintas pruebas, se ha configurado la calidad de servicio fijando el parámetro *Latency* a 5000 μs , 10000 μs y 15000 μs , en las dos conexiones RFCOMM y en los siguientes escenarios de prueba, esquematizados en la figura 5.37:

- Escenario 1: El esclavo E1 transmite datos al maestro, mientras que el maestro lo hace al esclavo E2.

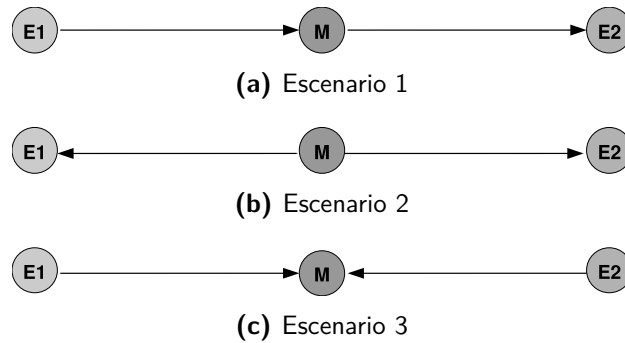


Figura 5.37: Escenarios de prueba en una *piconet* de 3 dispositivos

- Escenario 2: El maestro transmite a ambos esclavos simultáneamente.
- Escenario 3: Ambos esclavos transmiten datos, a la vez, al maestro.

En lo que concierne al periodo de transmisión, hay que indicar que entre iteraciones consecutivas se espera un tiempo aleatorio T , comprendido entre 90 y 110 ms, con el fin de que los envíos de datos de cada uno de los procesos, cliente o servidor, no estén completamente sincronizados.

A continuación se procede a analizar el resultado de las medidas efectuadas en relación a las cotas propuestas. Si bien hay que decir que, a efecto de reducir el número de experimentos mostrados, en esta memoria sólo se incluyen aquellas pruebas que emplean paquetes DH. No obstante, con paquetes DM se obtiene un comportamiento similar.

Una primera evaluación de los resultados condujo a concluir que, en el sentido de transmisión maestro-esclavo, el incremento que presenta el retardo medio medido es fijo e independiente del intervalo de *poll* configurado. Este *offset* resultó ser de 5 ms, que corresponde al tiempo medio de espera, $T_{poll}/2$, para un T_{poll} de 10 ms. Esto parece indicar que, en presencia de dos (o más) esclavos y cuando transmite el maestro, el dispositivo fija el valor real del intervalo de *poll*

al mínimo posible. Esta apreciación ha sido contemplada de forma que la cota mínima teórica representada para las conexiones en el que transmite el maestro en las figuras del primer y segundo escenario es la misma en todos los casos. Es decir, el *offset* sumado en la ecuación 5.9 en vez de ser $T_{poll}/2$, se ha considerado constante: $8 \cdot T_S$ (ó 5 ms).

Cuando el esclavo es el que transmite, la cota mínima $\overline{t_{R_{E_{min}}}}(N)$ sí depende del valor de $T_{poll}/2$. No obstante, se ha observado que existe un desplazamiento adicional de 2 ms, *offset* también presente en la *piconet* de un solo esclavo, que resultó ser independiente del valor del intervalo de *poll* configurado. Por este motivo, para los casos en los que el esclavo transmite al maestro, se ha representado, además de la cota inferior $\overline{t_{R_{E_{min}}}}(N)$, la curva resultante de desplazar $\overline{t_{R_{E_{min}}}}(N)$ un tiempo de 2 ms.

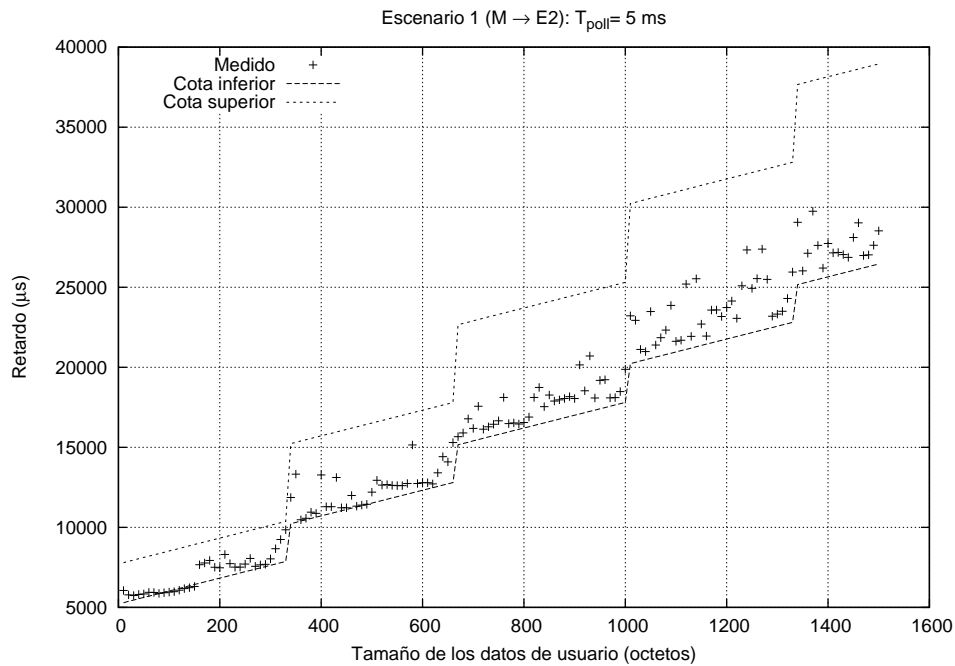
Las figuras 5.38–5.46 representan los retardos medidos para distintos valores del parámetro *Latency* y cada uno de los escenarios indicados al comienzo de la sección, junto con las cotas mínimas y máximas dadas por las ecuaciones 5.9 y 5.10, para la transmisión maestro-esclavo, y por las ecuaciones 5.12 y 5.13, en el sentido de transmisión esclavo-maestro, teniendo en cuenta las aclaraciones realizadas en relación con las cotas mínimas. En estas figuras puede comprobarse que, en todas las transmisiones esclavo-maestro, el retardo queda perfectamente delimitado inferiormente al sumar 2 ms al resultado de la ecuación 5.12 y que, para el sentido maestro-esclavo, la cota inferior es independiente del intervalo de *poll* configurado, como también se muestra en la figura 5.47, para $T_{poll} = 20 \text{ ms}$ y las conexiones maestro-esclavo en los Escenarios 1 y 2.

Una observación de gran relevancia es que en las gráficas de la figura 5.39, correspondiente al escenario en el que el maestro transmite a los dos esclavos, puede verse que para ambas conexiones la cota es excedida (en el caso (b) incluso aún más que en el caso (a) de la figura 5.39). Sin embargo, en la gráfica de la figura 5.38 del escenario 1, puede comprobarse que el retardo medido para la transmisión de datos a uno de los esclavos resulta ser muy inferior al máximo previsto. La explicación que cabe argüir es que, en el escenario 2, las dos conexiones RFCOMM se multiplexan sobre un mismo dispositivo *hardware*, compartiéndose por tanto, los recursos de memoria RAM (*Random Access Memory*). Este argumento justifica que la situación sea más crítica cuánto mayor es el tamaño de los datos que se transmiten a los esclavos. De hecho, en las especificaciones del fabricante CSR [CSR] se indica que el máximo número de conexiones que pueden mantenerse, dependerá precisamente de la memoria RAM disponible. Además, un factor que agrava el colapso de la memoria RAM es la coincidencia en las peticiones de transmisión que, para ambas conexiones RFCOMM están espaciadas un periodo de transmisión de 100 ms con sólo pequeñas variaciones de ± 10 ms. Al respecto, es conveniente señalar que esta sincronización entre los periodos de transmisión es más acusada debido al propio procedimiento de medida, ya que los procesos clientes y servidores tienen la misma referencia temporal. Como justificación, en la figura 5.48 se representa el retardo obtenido al repetir la medida para el escenario 2 y distintos periodos de transmisión: $T_1 = 100\text{ ms} \pm 10\text{ ms}$ y $T_2 = 101\text{ ms} \pm 10\text{ ms}$, ya que así se evita completamente la sincronía de los envíos y, por tanto, el colapso de la memoria RAM.

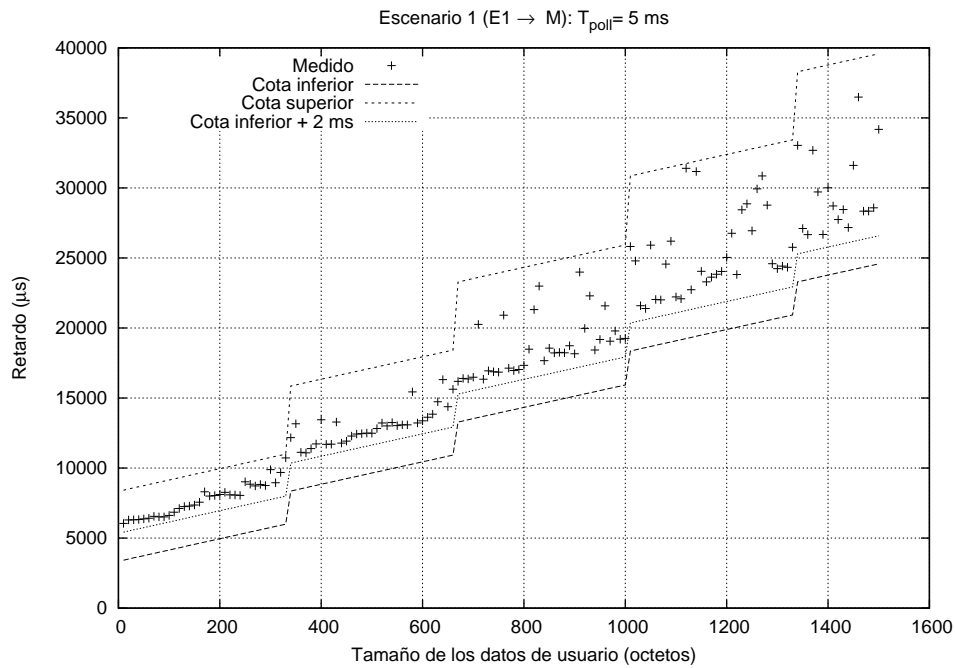
Los resultados obtenidos también permiten validar las cotas mínima y

máxima propuestas para la componente $t_{ACK}(N)$, ya que si se considera que cada una de las posibilidades ilustradas en las figuras 5.33 y 5.34 son equiprobables, es decir, si el 50% de las veces la separación entre fragmentos consecutivos es de 8 *slots* ($8 \cdot T_S$) y en el otro 50%, lo es de 12 *slots* ($12 \cdot T_S$), el retardo obtenido al promediar las 1000 repeticiones se debería ajustar, tal y como sucede, según puede observarse en la figura 5.48, al retardo teórico calculado a partir de la ecuación 5.9, que define la cota mínima $\overline{t_{R_{M_{min}}}}(N)$, suponiendo que la separación entre fragmentos es de 10 *slots* ($10 \cdot T_S$) (que es el promedio de $8 \cdot T_S$ y de $12 \cdot T_S$).

5.3 Efecto de la configuración de la calidad de servicio

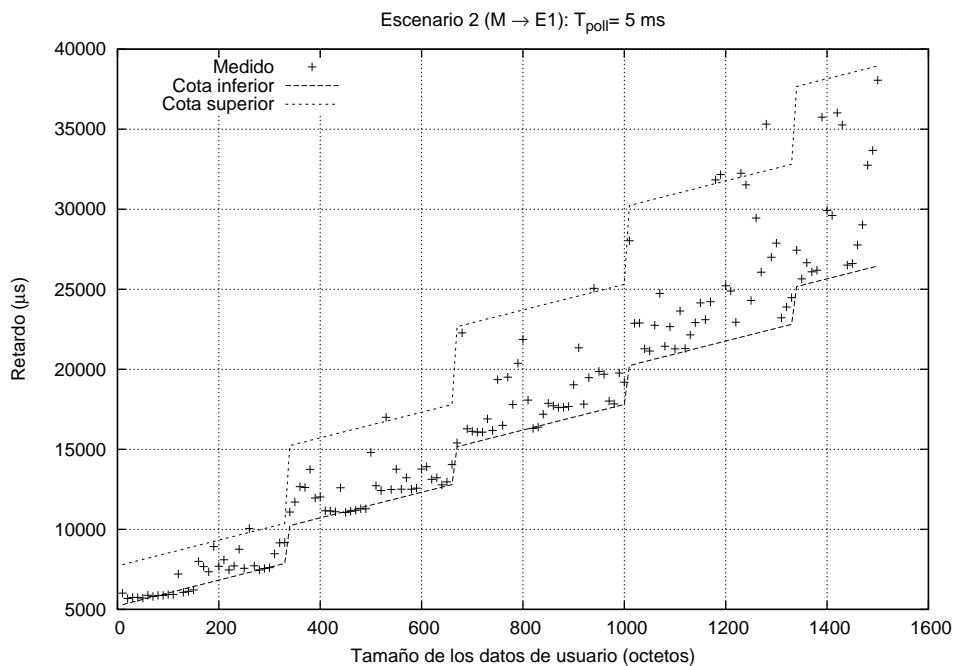


(a) Transmisión del maestro al esclavo E2

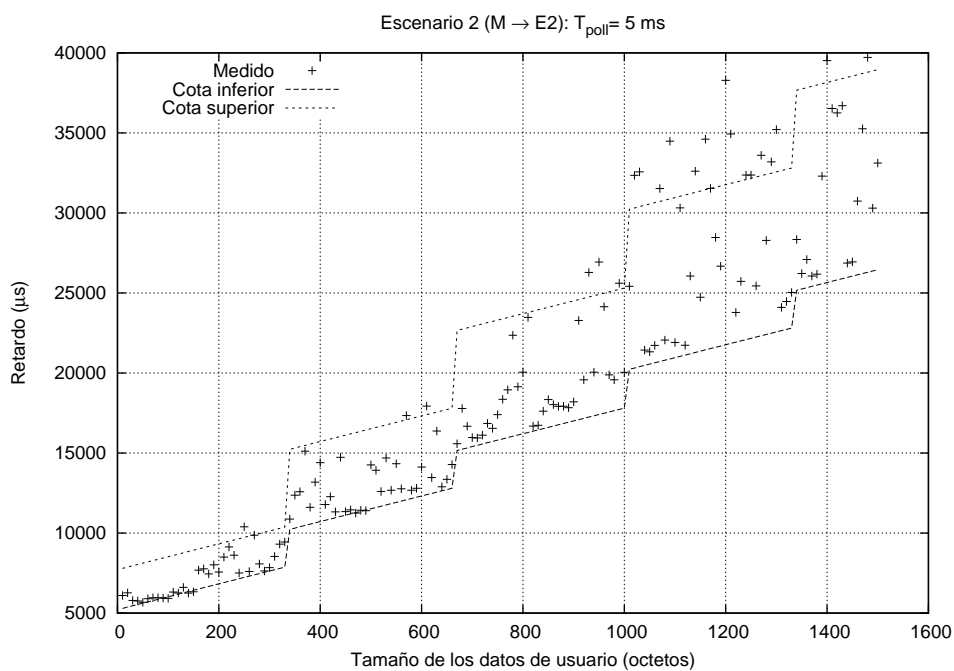


(b) Transmisión del esclavo E1 al maestro

Figura 5.38: Escenario 1: Cotas mínimas y máximas, dadas por las ecuaciones 5.9 y 5.10 (subfigura (a)), y 5.12 y 5.13 (subfigura (b)), y retardo de transmisión medido en cada conexión, con paquetes DH y un valor del parámetro T_{poll} de 5 ms



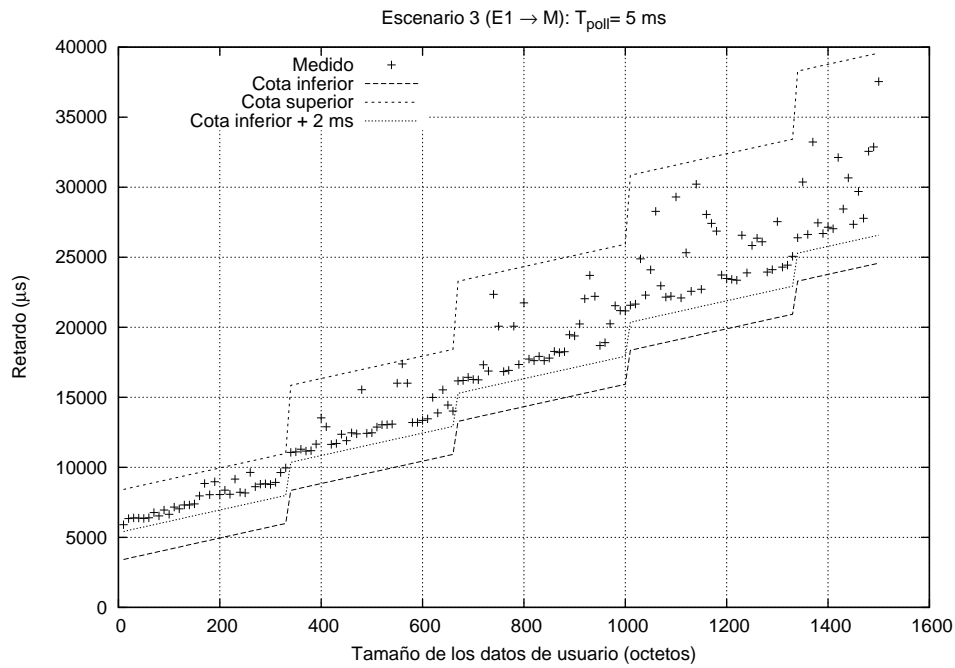
(a) Transmisión del maestro al esclavo E1



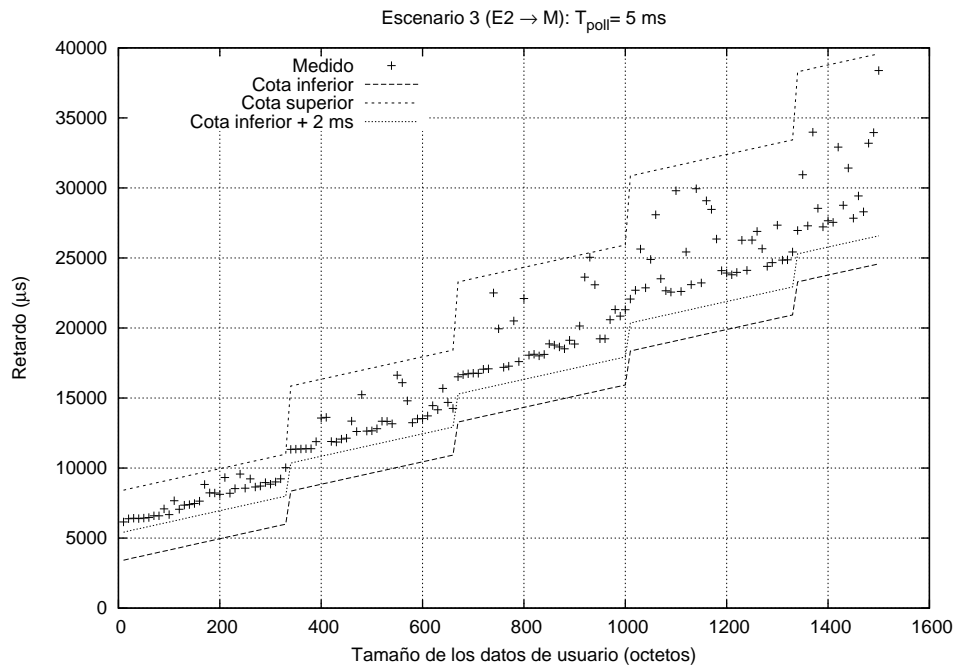
(b) Transmisión del maestro al esclavo E2

Figura 5.39: Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.9 y 5.10, y retardo de transmisión medido en cada conexión, con paquetes DH y un valor del parámetro T_{poll} de 5 ms

5.3 Efecto de la configuración de la calidad de servicio

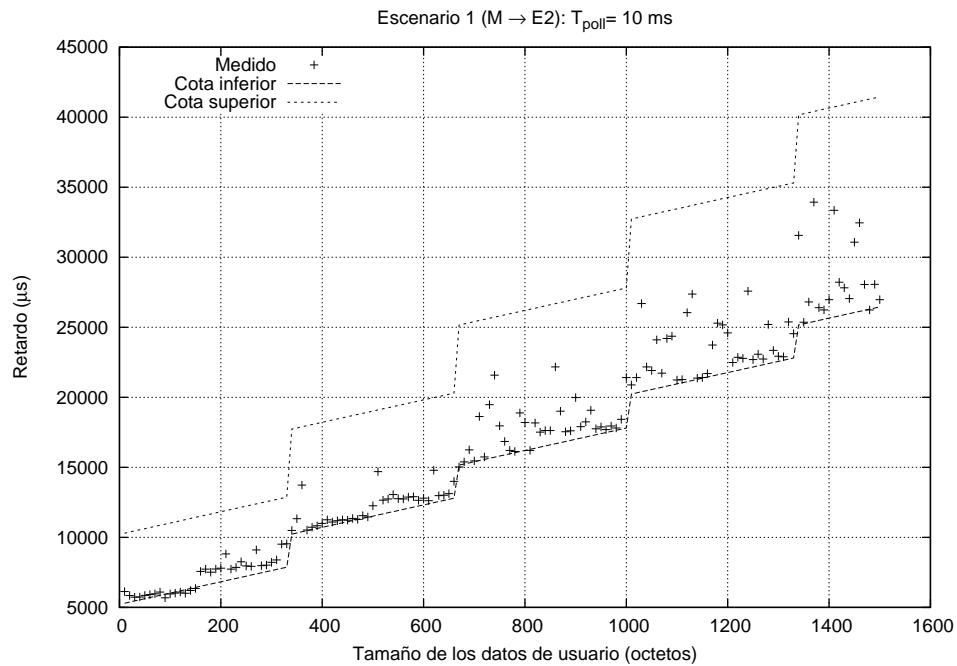


(a) Transmisión del esclavo E1 al maestro

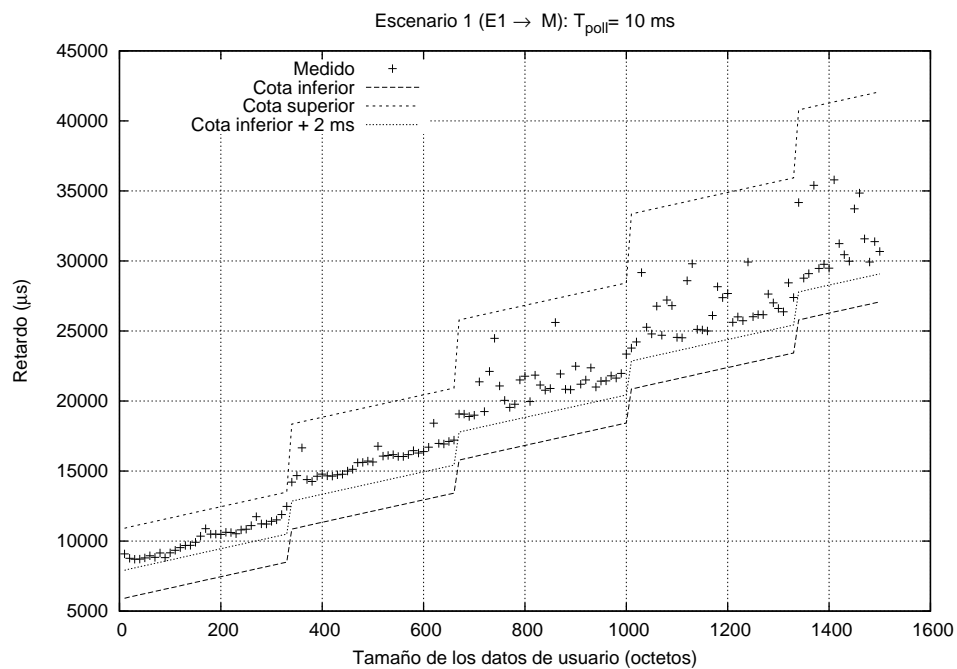


(b) Transmisión del esclavo E2 al maestro

Figura 5.40: Escenario 3: Cotas mínima y máxima, dadas por las ecuaciones 5.12 y 5.13, y retardo de transmisión medido en cada conexión, con paquetes DH y un valor del parámetro T_{poll} de 5 ms



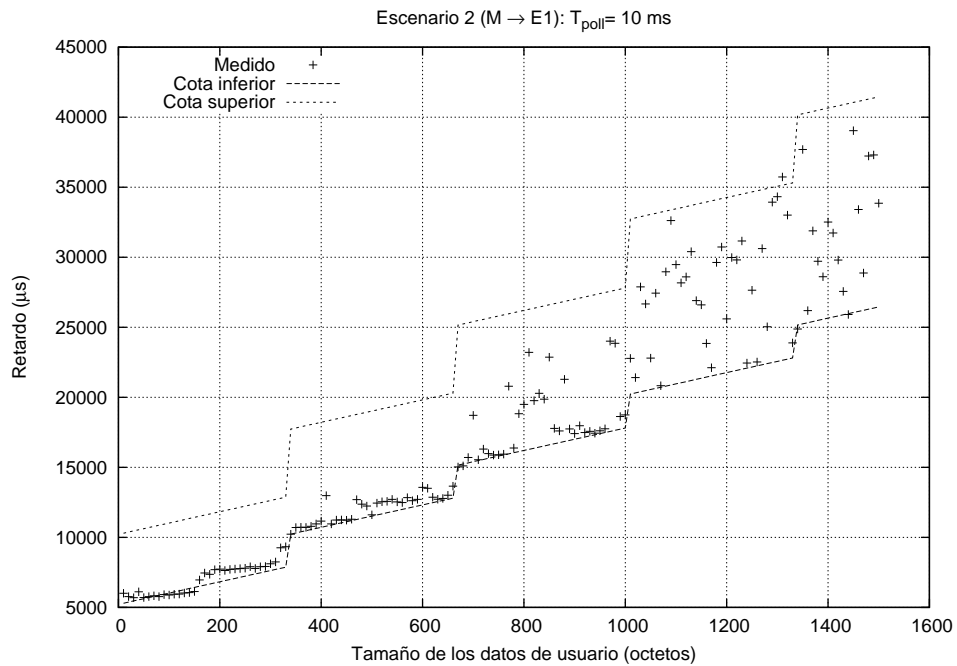
(a) Transmisión del maestro al esclavo E2



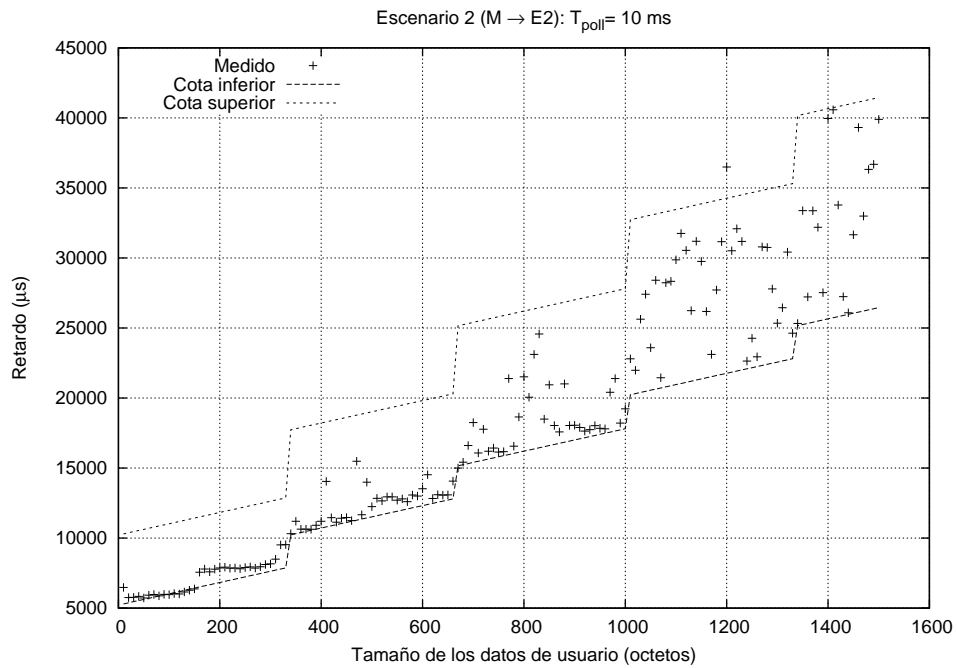
(b) Transmisión del esclavo E1 al maestro

Figura 5.41: Escenario 1: Cotas mínimas y máximas, dadas por las ecuaciones 5.9 y 5.10 (subfigura (a)), y 5.12 y 5.13 (subfigura (b)), y retardo de transmisión medido en cada conexión, con paquetes DH y un valor del parámetro T_{poll} de 10 ms

5.3 Efecto de la configuración de la calidad de servicio

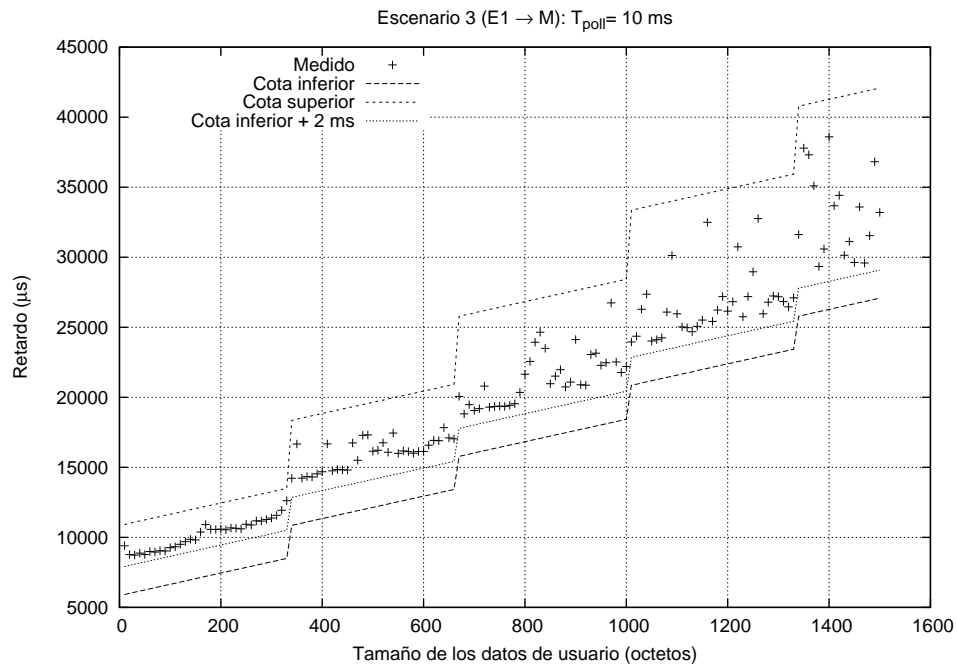


(a) Transmisión del maestro al esclavo E1

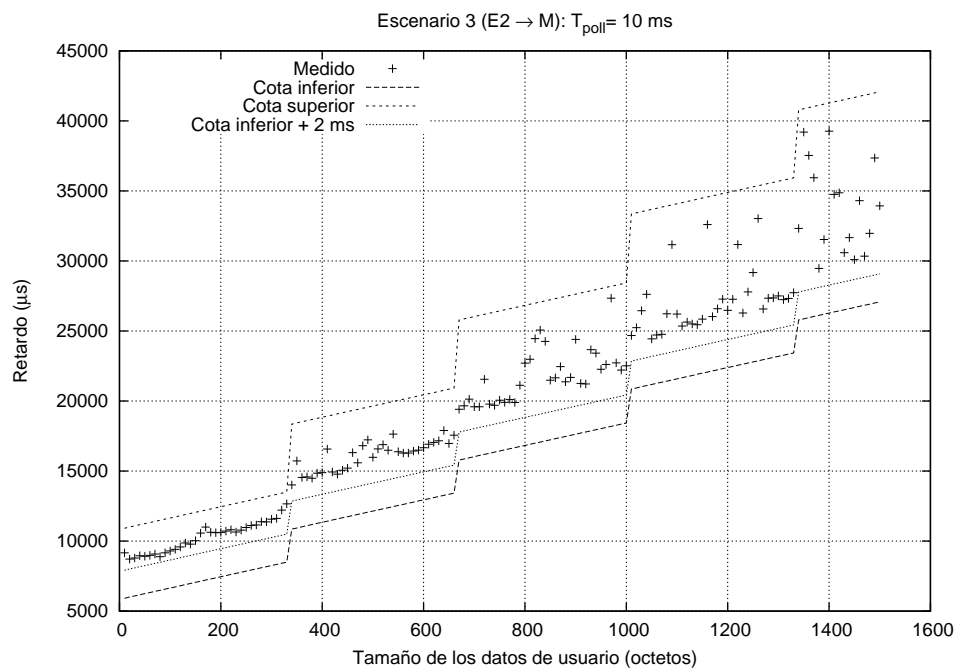


(b) Transmisión del maestro al esclavo E2

Figura 5.42: Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.9 y 5.10, y retardo de transmisión medido en cada conexión, con paquetes DH y un valor del parámetro T_{poll} de 10 ms



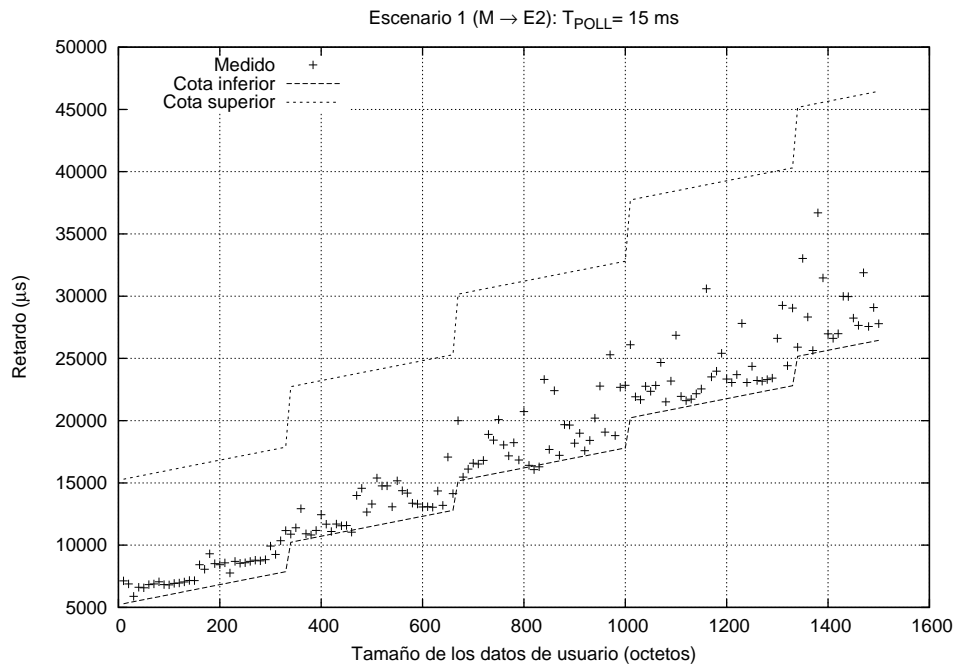
(a) Transmisión del esclavo E1 al maestro



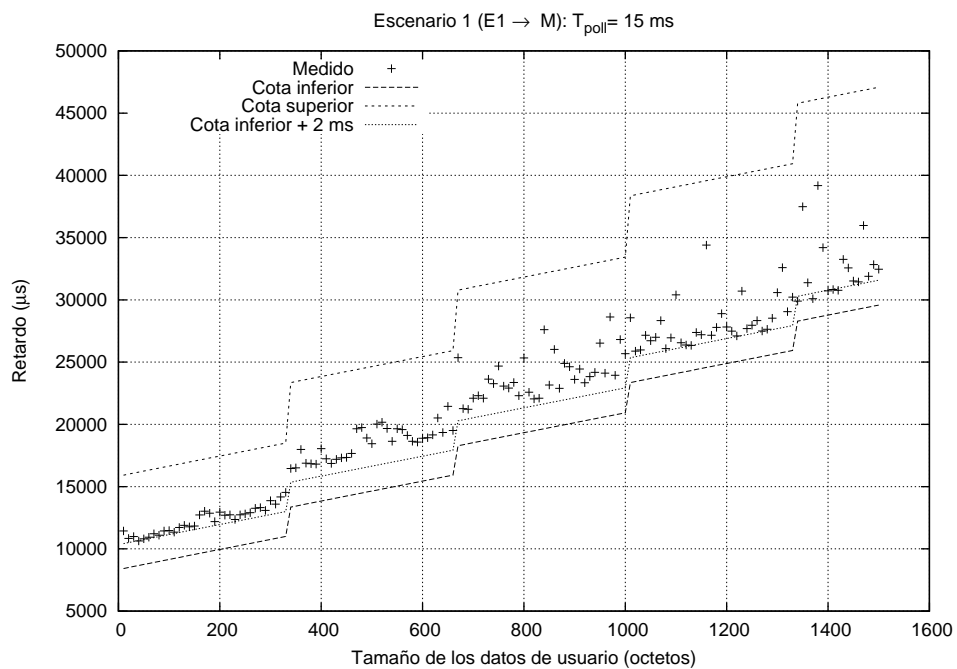
(b) Transmisión del esclavo E2 al maestro

Figura 5.43: Escenario 3: Cotas mínima y máxima, dadas por las ecuaciones 5.12 y 5.13, y retardo de transmisión medido en cada conexión, con paquetes DH y un valor del parámetro T_{poll} de 10 ms

5.3 Efecto de la configuración de la calidad de servicio

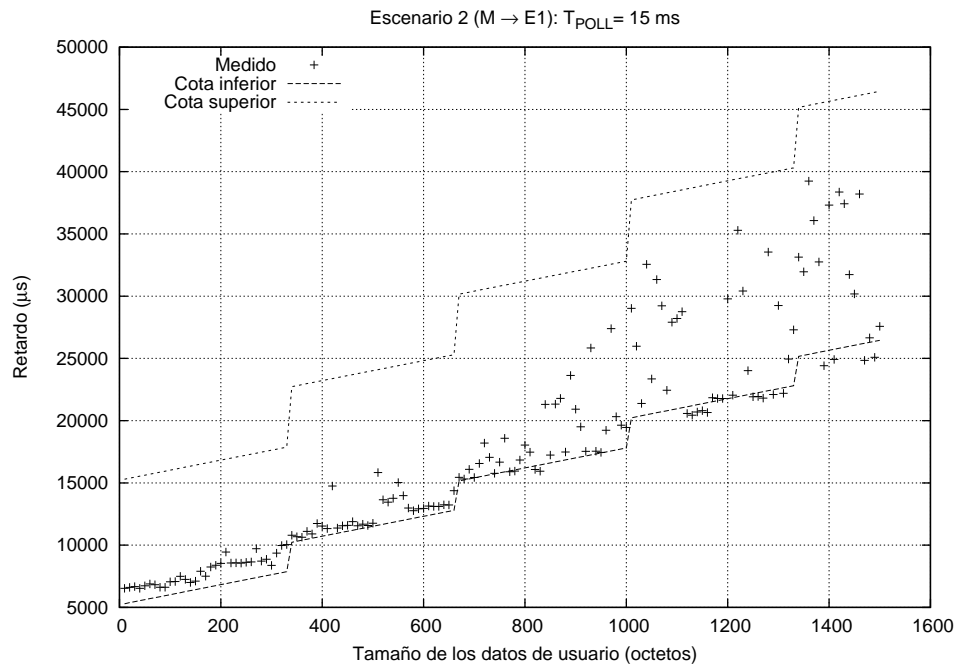


(a) Transmisión del maestro al esclavo E2

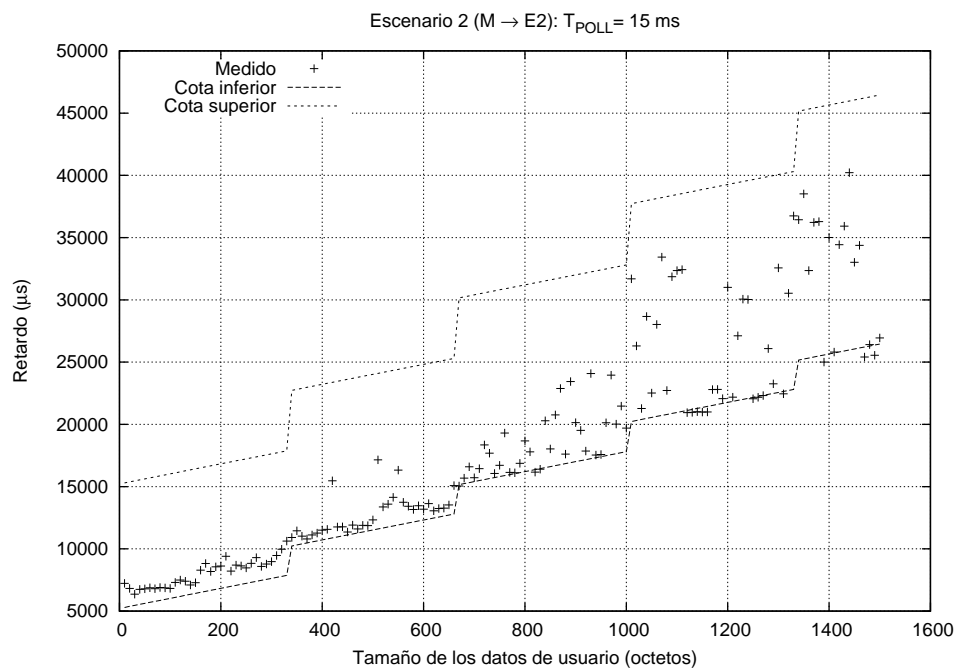


(b) Transmisión del esclavo E1 al maestro

Figura 5.44: Escenario 1: Cotas mínimas y máximas, dadas por las ecuaciones 5.9 y 5.10 (subfigura (a)), y 5.12 y 5.13 (subfigura (b)), y retardo de transmisión medido en cada conexión, con paquetes DH y un valor del parámetro T_{poll} de 15 ms



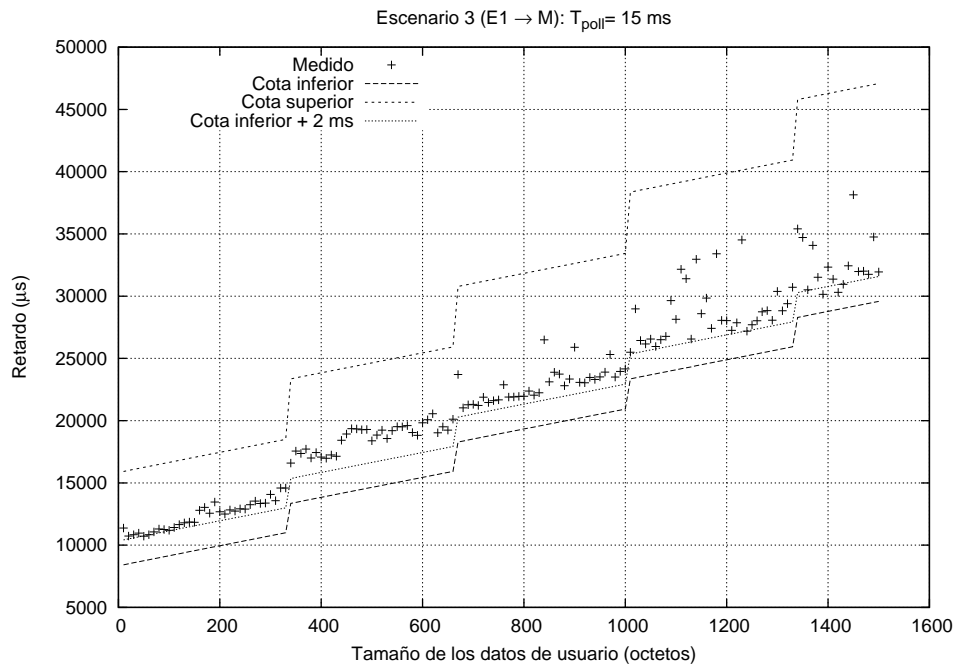
(a) Transmisión del maestro al esclavo E1



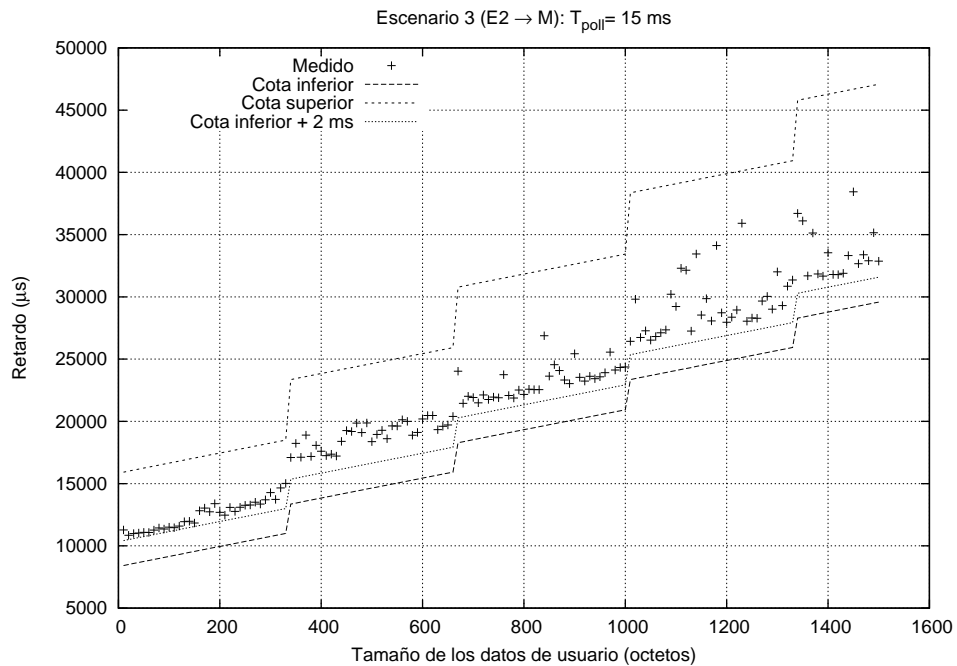
(b) Transmisión del maestro al esclavo E2

Figura 5.45: Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.9 y 5.10, y retardo de transmisión medido en cada conexión, con paquetes DH y un valor del parámetro T_{poll} de 15 ms

5.3 Efecto de la configuración de la calidad de servicio

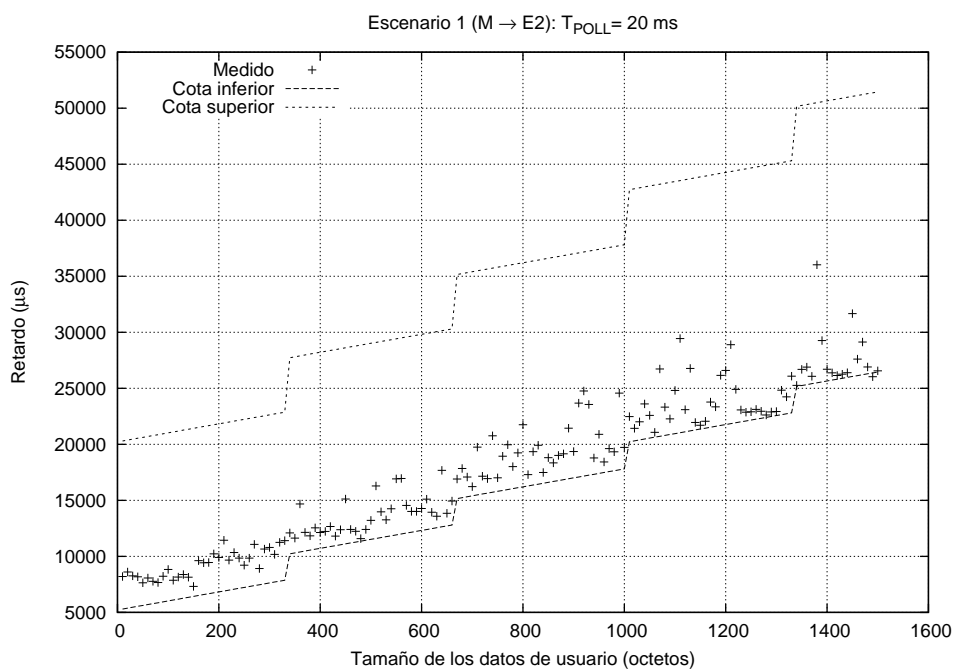


(a) Transmisión Esclavo 1 - Maestro

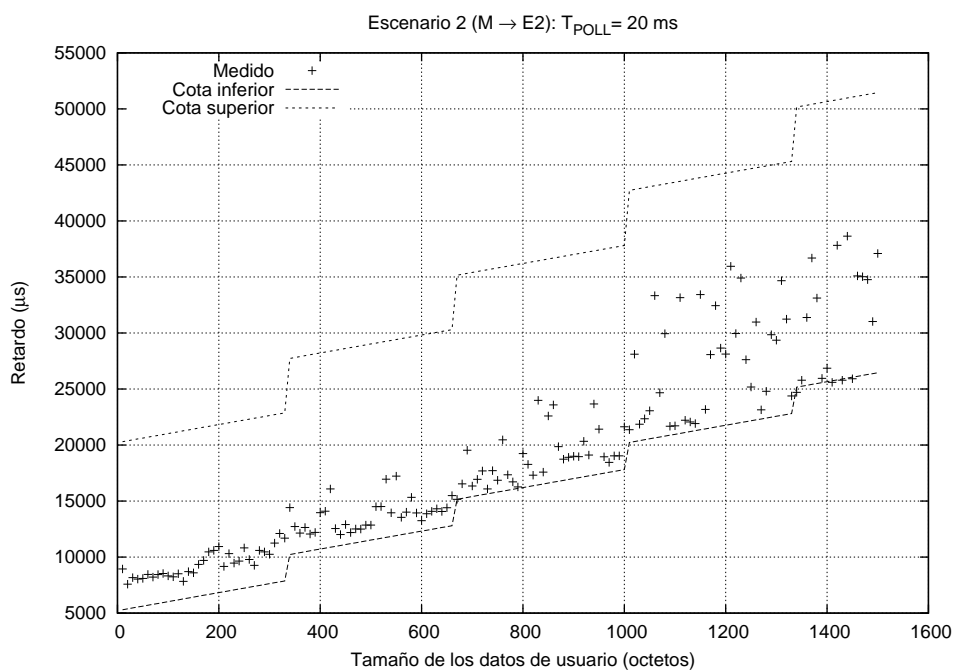


(b) Transmisión Esclavo 2 - Maestro

Figura 5.46: Escenario 3: Cotas mínima y máxima, dadas por las ecuaciones 5.12 y 5.13, y retardo de transmisión medido en cada conexión, con paquetes DH y un valor del parámetro T_{poll} de 15 ms



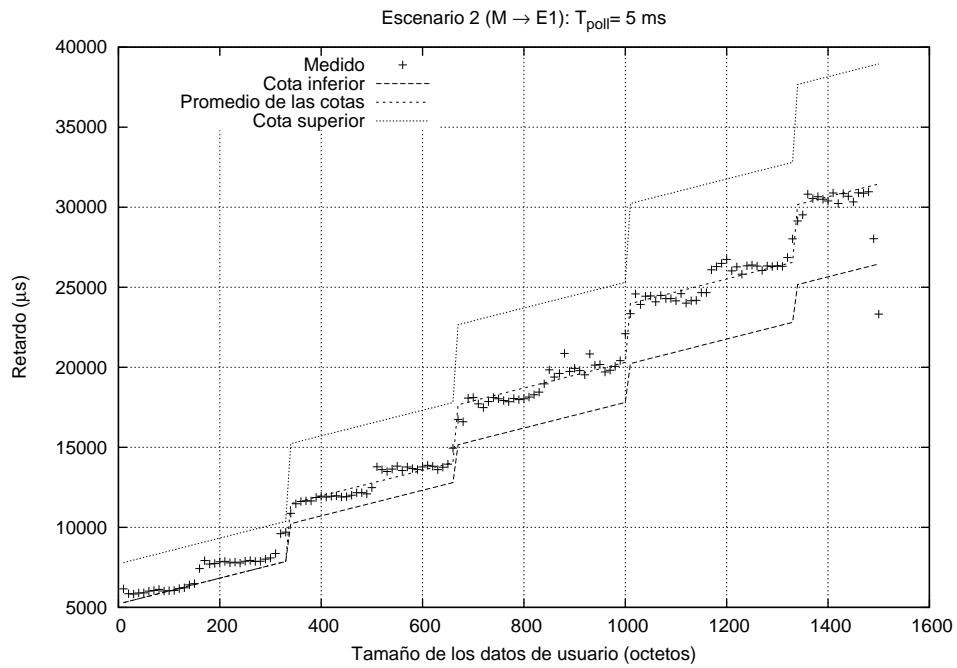
(a) Escenario 1: Transmisión del maestro al esclavo E2



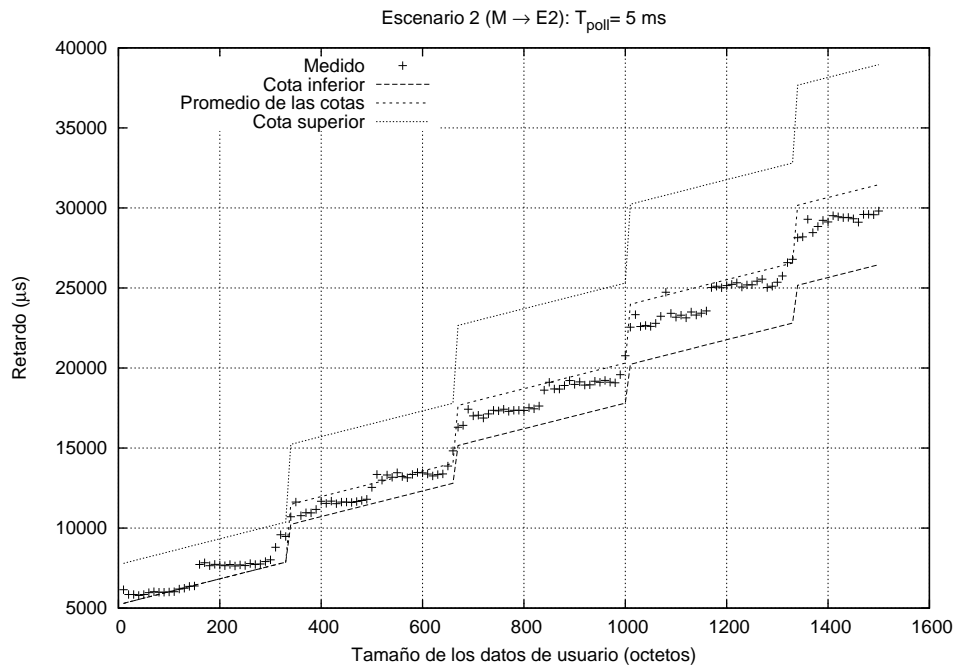
(b) Escenario 2: Transmisión del maestro al esclavo E2

Figura 5.47: Escenarios 1 y 2: Cotas mínima y máxima, dadas por las ecuaciones 5.9 y 5.10 y retardo de transmisión medido en la conexión maestro-esclavo, con paquetes DH y un valor del parámetro T_{poll} de 20 ms

5.3 Efecto de la configuración de la calidad de servicio



(a) Transmisión del maestro al esclavo E1: $T = 101$ ms



(b) Transmisión del maestro al esclavo E2: $T = 100$ ms

Figura 5.48: Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.9 y 5.10, y retardo de transmisión medido en cada conexión, con paquetes DH, un valor del parámetro T_{poll} de 5 ms y periodos distintos

5.3.3. Generalización del modelo para *piconets* de más de dos dispositivos

En este apartado se generalizan las cotas propuestas para la *piconet* de tres dispositivos. Al igual que se hizo en este caso, se va a comenzar redefiniendo la componente del retardo $t_{ACK}(N)$, teniendo en cuenta que el maestro va a direccionar a todos los esclavos siguiendo un algoritmo de planificación RR, pudiendo darse las dos situaciones siguientes:

- Las transmisiones a los esclavos no se sincronizan en el tiempo: Esto es, el maestro sólo tiene datos pendientes de transmisión para un esclavo. Así, el tiempo mínimo que el maestro tardaría en direccionar a este esclavo se computaría como el número de *slots* ocupados por la transmisión al esclavo en cuestión (1, 3 ó 5 *slots*), 1 *slot* de confirmación del paquete recibido y 2 *slots* por cada uno de los restantes esclavos (uno de direccionamiento y otro de respuesta al maestro). Así, la cota mínima de la componente $t_{ACK}(N)$, $t_{ACK_{min}}(N, n_e)$, tendría que ser redefinida de acuerdo con la siguiente ecuación:

$$t_{ACK_{min}}(N, n_e) = \begin{cases} 0 & N = 0 \\ 2 \cdot n_e \cdot T_s & 0 < N \leq L_1 \\ (4 + 2 \cdot (n_e - 1)) \cdot T_s & L_1 < N \leq L_3 \\ (6 + 2 \cdot (n_e - 1)) \cdot T_s & L_3 < N \leq L_5 \\ (6 + 2 \cdot (n_e - 1)) \cdot T_s \cdot \left\lfloor \frac{N}{L_5} \right\rfloor + & N > L_5 \\ + t_{ACK_{min}}(N \bmod L_5, n_e) & \end{cases} \quad (5.14)$$

donde n_e representa el número de esclavos participantes, en modo activo (*ACTIVE*), en la *piconet*.

- Todas las peticiones de datos se sincronizan en el tiempo: En este caso, el maestro tiene datos pendientes de transmisión para todos los esclavos, de forma que las transmisiones de datos a los esclavos se sucederán en secuencia. El máximo tiempo que un determinado esclavo tardará en ser direccionado se alcanza cuando el maestro ha de transmitir a los esclavos restantes paquetes de 5 *slots*. Por consiguiente, la cota máxima de la componente $t_{ACK}(N)$, $t_{ACK_{max}}(N, n_e)$, vendrá dada por la siguiente ecuación:

$$t_{ACK_{max}}(N, n_e) = \begin{cases} 0 & N = 0 \\ (6 \cdot (n_e - 1) + 2) \cdot T_s & 0 < N \leq L_1 \\ (6 \cdot (n_e - 1) + 4) \cdot T_s & L_1 < N \leq L_3 \\ 6 \cdot n_e \cdot T_s & L_3 < N \leq L_5 \\ (6 \cdot n_e \cdot T_s) \cdot \left\lfloor \frac{N}{L_5} \right\rfloor + \\ + t_{ACK_{max}}(N \bmod L_5, n_e) & N > L_5 \end{cases} \quad (5.15)$$

Al igual que en el caso particular de la *piconet* de tres dispositivos, las ecuaciones $t_{ACK_{min}}(N, n_e)$ y $t_{ACK_{max}}(N, n_e)$ definidas para el sentido de transmisión maestro-esclavo son también válidas para el caso en el que sea el esclavo el que transmite.

Adicionalmente, debido a la inclusión del parámetro n_e en la componente del retardo $t_{ACK}(N)$, se hace necesario redefinir la ecuación que define $t_{TX}(N)$, tal y como se indica a continuación:

$$t_{TX}(N, n_e) = \begin{cases} 0 & N = 0 \\ n_b(N) \cdot t_b & 0 < N \leq L_5 \\ t_{ACK}(L_5, n_e) \cdot \left\lfloor \frac{N}{L_5} \right\rfloor + & N > L_5 \\ +t_{TX}(N \bmod L_5, n_e) & \end{cases} \quad (5.16)$$

Por último, queda incorporar las ecuaciones $t_{ACK_{min}}(N, n_e)$ y $t_{ACK_{max}}(N, n_e)$ a las expresiones del retardo de transmisión propuestas para el perfil SPP en el caso de una *piconet* de n_e dispositivos: $t_{RM}(N)$ y $t_{RE}(N)$. Igualmente, en sendas ecuaciones se va a introducir un *offset* distinto para cada sentido de transmisión, según lo que ha sido inferido a partir de las medidas realizadas para el caso particular de la *piconet* de tres dispositivos:

Sentido de transmisión maestro-esclavo: La cota mínima del retardo de transmisión utilizando el perfil SPP, $\overline{t_{RM_{min}}}(N, n_e)$, vendría dada por la siguiente ecuación:

$$\begin{aligned} \overline{t_{RM_{min}}}(N, n_e) = 8 \times T_S + n_{fnf_R}(N) \cdot t_{ACK_{min}}(L_R + O_R(L_R) + C_L, n_e) + \\ + t_{TX}(L_{ff_R} + O_R(L_{ff_R}) + C_L, n_e) \end{aligned} \quad (5.17)$$

Obsérvese que, de acuerdo con lo que se ha constatado empíricamente sobre las transmisiones maestro-esclavo en el apartado 5.3.2, la cota mínima no depende del valor del intervalo de *poll* configurado. De hecho, el desplazamiento de 8 *slots* ($8 \cdot T_S$) corresponde a un tiempo de 5 ms, es decir, $T_{poll}/2$ para $T_{poll} = 10 \text{ ms}$.

Sin embargo, la cota superior, $t_{R_{Mmax}}(N, n_e)$, que sí depende del valor de T_{poll} , respondería a la siguiente ecuación:

$$t_{R_{Mmax}}(N, n_e) = \Delta_{poll}(n_e) + n_{fnf_R}(N) \cdot t_{ACK_{max}}(L_R + O_R(L_R) + C_L, n_e) + t_{TX}(L_{ff_R} + O_R(L_{ff_R}) + C_L, n_e) \quad (5.18)$$

donde $\Delta_{poll}(n_e)$, tiene en cuenta que el intervalo de *poll* configurado pueda o no garantizarse, de acuerdo con la siguiente expresión:

$$\Delta_{poll}(n_e) = \max(6 \cdot n_e \cdot T_S, T_{poll}) \quad (5.19)$$

La fórmula 5.19, para un valor de T_{poll} que no pueda garantizarse, fija el incremento del retardo en un valor dependiente del número de esclavos: $6 \cdot n_e \cdot T_S$.

Sentido de transmisión esclavo-maestro: En este caso, la generalización de la cota mínima del retardo de transmisión para el perfil SPP y una *piconet* de tres dispositivos, $\overline{t_{R_{Emin}}}(N)$, dada por la ecuación 5.12 conduce a la siguiente expresión:

$$\overline{t_{R_{Emin}}}(N, n_e) = T_{poll}/2 + T_S + n_{fnf_R}(N) \cdot t_{ACK_{min}}(L_R + O_R(L_R) + C_L, n_e) + t_{TX}(L_{ff_R} + O_R(L_{ff_R}) + C_L, n_e) \quad (5.20)$$

Como puede observarse, cuando el esclavo es el que transmite, sí existe dependencia respecto del valor de $T_{poll}/2$.

Asimismo, la cota superior $t_{R_{E_{max}}}(N)$ definida mediante la ecuación 5.13, al generalizarla para una piconet de n_e esclavos viene dada por la siguiente fórmula:

$$t_{R_{E_{max}}}(N, n_e) = t_{R_{M_{max}}}(N, n_e) + T_S \quad (5.21)$$

5.3.4. Medidas para una *piconet* de cuatro dispositivos

El objetivo es mostrar si las cotas propuestas en el apartado anterior permiten modelar el comportamiento de una *piconet* de cuatro dispositivos. Es decir, si las cotas $\overline{t_{R_{E_{min}}}}(N, n_e)$ y $t_{R_{E_{max}}}(N, n_e)$, así como $\overline{t_{R_{M_{min}}}}(N, n_e)$ y $t_{R_{M_{max}}}(N, n_e)$, particularizadas para $n_e = 3$, se satisfacen, respectivamente, en el sentido de transmisión esclavo-maestro y maestro-esclavo.

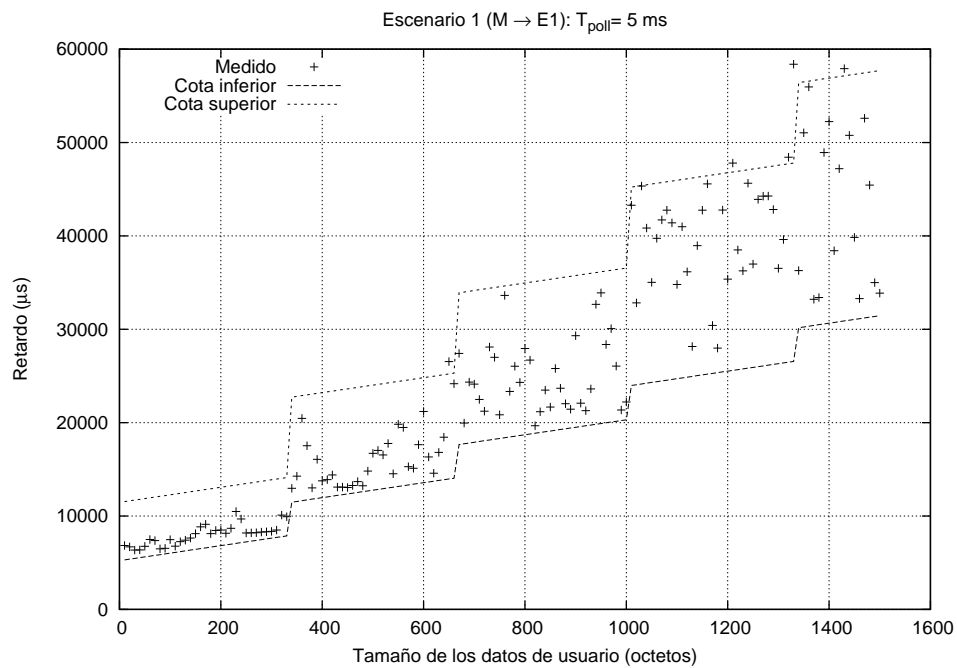
Las pruebas se han llevado a cabo estableciendo tres conexiones RFCOMM, con un periodo de transmisión variable de $100\text{ ms} \pm 30\text{ ms}$ y configurando, en cada conexión, la calidad de servicio con el parámetro *Latency* a $5000\ \mu\text{s}$, $10000\ \mu\text{s}$ y $15000\ \mu\text{s}$. En este caso se han considerado los siguientes escenarios de prueba:

- Escenario 1: El maestro transmite a todos los esclavos (E1, E2 y E3) simultáneamente.
- Escenario 2: Todos los esclavos transmiten datos, a la vez, al maestro.

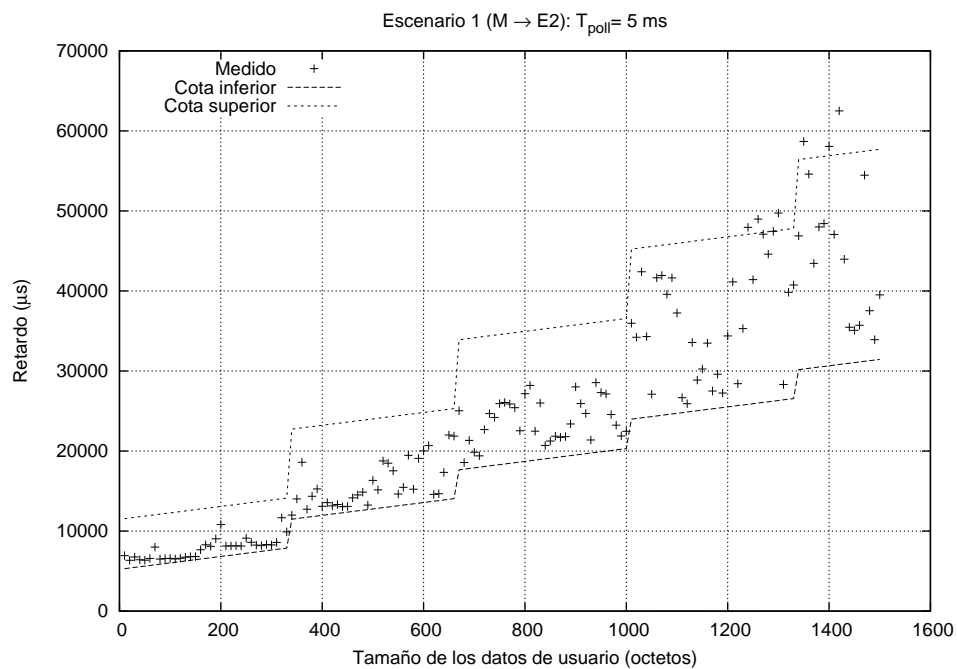
Nótese que, respecto a la *piconet* de tres dispositivos, se ha aumentado en 20 ms el intervalo en el que puede tomar valores el periodo de transmisión. Así, se ha pretendido disminuir la probabilidad de colisión entre los procesos que transmiten datos.

En las figuras 5.51, 5.52, 5.55, 5.56, 5.59 y 5.60 correspondientes al escenario 2, puede observarse que el retardo medido queda confinado por los límites inferior y superior obtenidos a partir de las ecuaciones 5.20 y 5.21, y que, además, es posible mejorar la cota inferior sumando el desplazamiento de 2 ms, que también se tuvo en cuenta en la *piconet* de tres dispositivos.

En cuanto a las gráficas 5.49, 5.50, 5.53, 5.54, 5.57 y 5.58, que presentan los resultados obtenidos para el escenario 1, hay que reseñar que, principalmente, para intervalos de *poll* de 5 ms, la cota superior es excedida. Puede apreciarse que la conexión con el esclavo E3 es la más perjudicada, ya que se alcanzan mayores retardos que en las otras dos conexiones, incluso a longitudes inferiores. Tal y como se explicó en el apartado 5.3.2, esto es debido a la multiplexación de los datos de tres conexiones RFCOMM sobre un mismo dispositivo (el maestro), efecto que no está contemplado en el modelo. Aunque esta hipótesis quedó probada para la *piconet* de dos esclavos, se ha optado por incluir las gráficas mostradas en la figura 5.61. El retardo medido, que se muestra en esta figura, corresponde al obtenido en el sentido maestro-esclavo en un tercer escenario en el que el maestro transmite a un esclavo, mientras que los otros dos esclavos envían datos al maestro, habiendo configurado un intervalo de *poll* de 5 ms (figura 5.61 (a)) y de 10 ms (figura 5.61 (b)) para todas las conexiones.



(a) Escenario 1: Transmisión del maestro al esclavo E1



(b) Escenario 1: Transmisión del maestro al esclavo E2

Figura 5.49: Cotas mínima y máxima, dadas por las ecuaciones 5.17 y 5.18, y retardo medido en las transmisiones del maestro a los esclavos E1 y E2, con paquetes DH y un valor del parámetro T_{poll} de 5 ms

5.3 Efecto de la configuración de la calidad de servicio

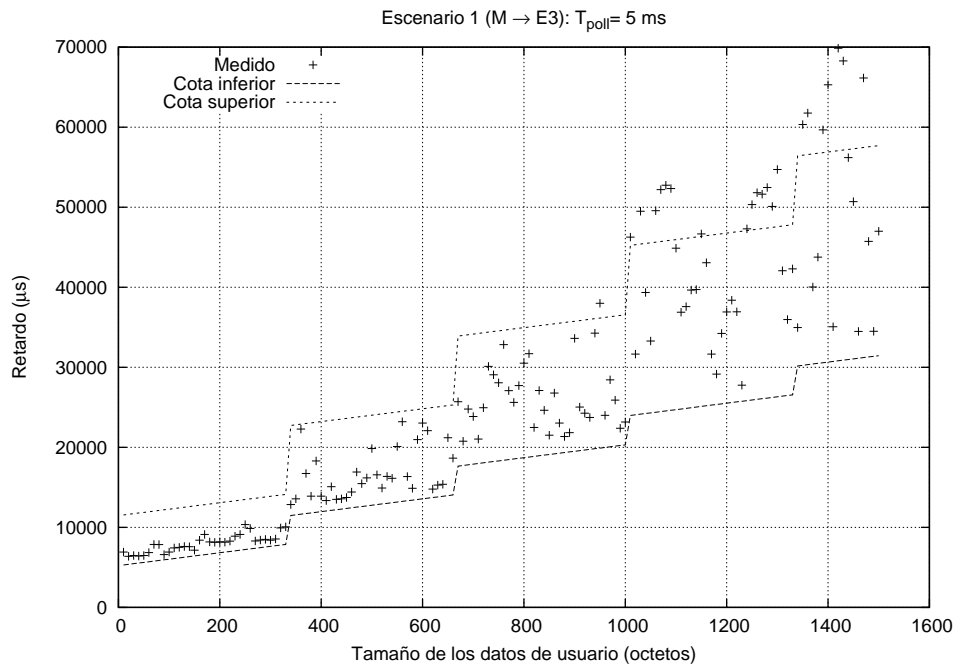


Figura 5.50: Escenario 1: Cotas mínima y máxima, dadas por las ecuaciones 5.17 y 5.18, y retardo medido en la transmisión del maestro al esclavo E3, con paquetes DH y un valor del parámetro T_{poll} de 5 ms

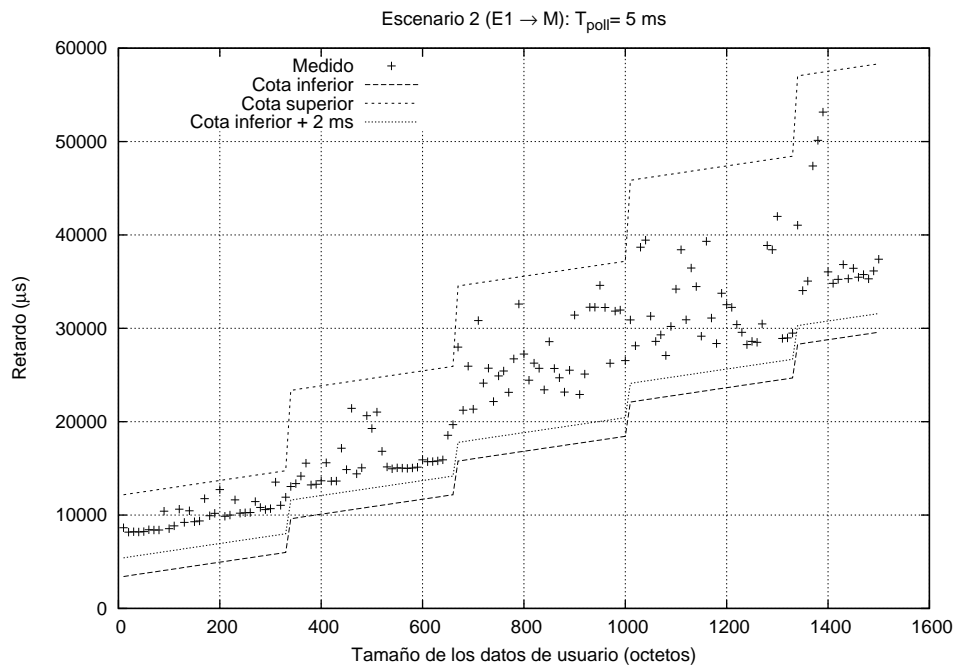
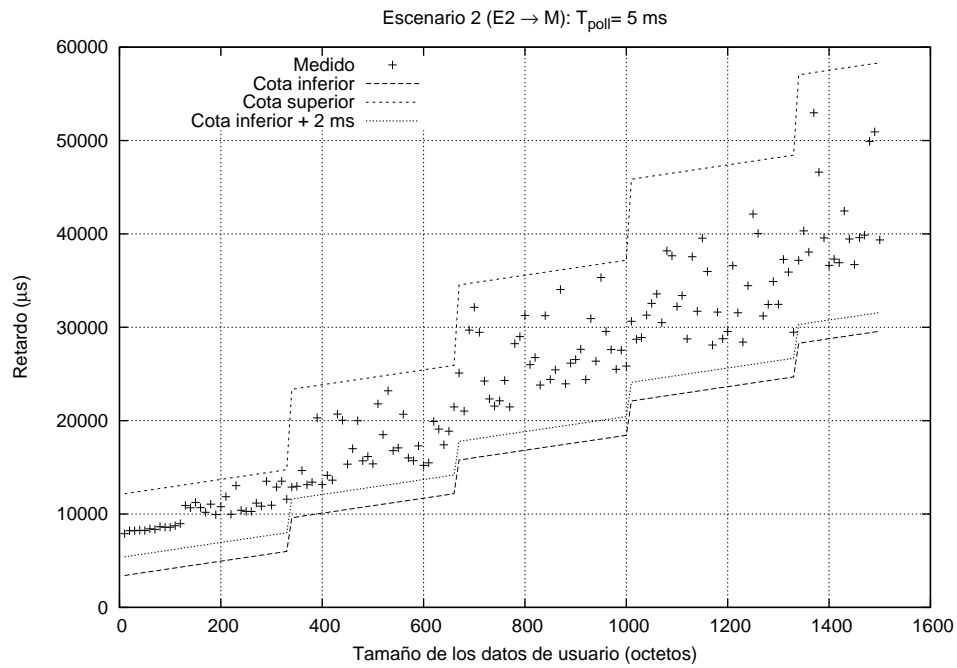
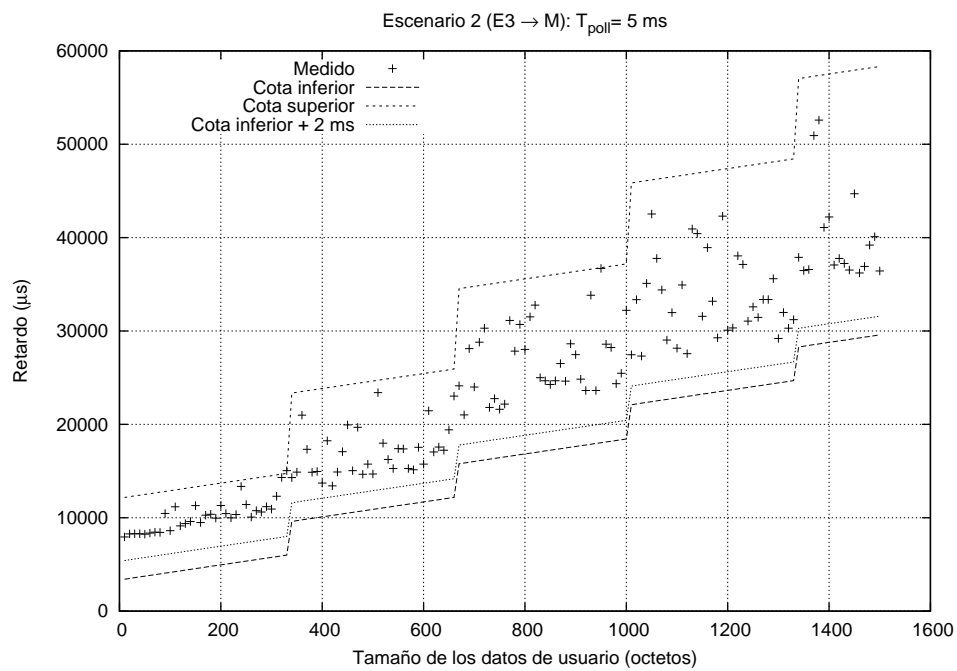


Figura 5.51: Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.20 y 5.21, y retardo medido en la transmisión del esclavo E1 al maestro, con paquetes DH y un valor del parámetro T_{poll} de 5 ms



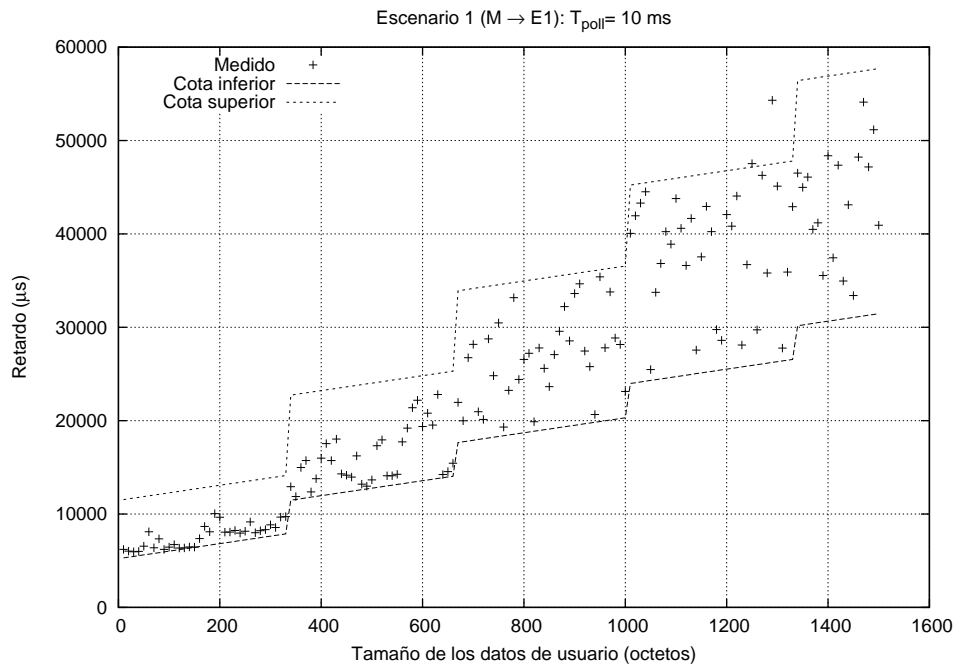
(a) Escenario 2: Transmisión del esclavo E2 al maestro



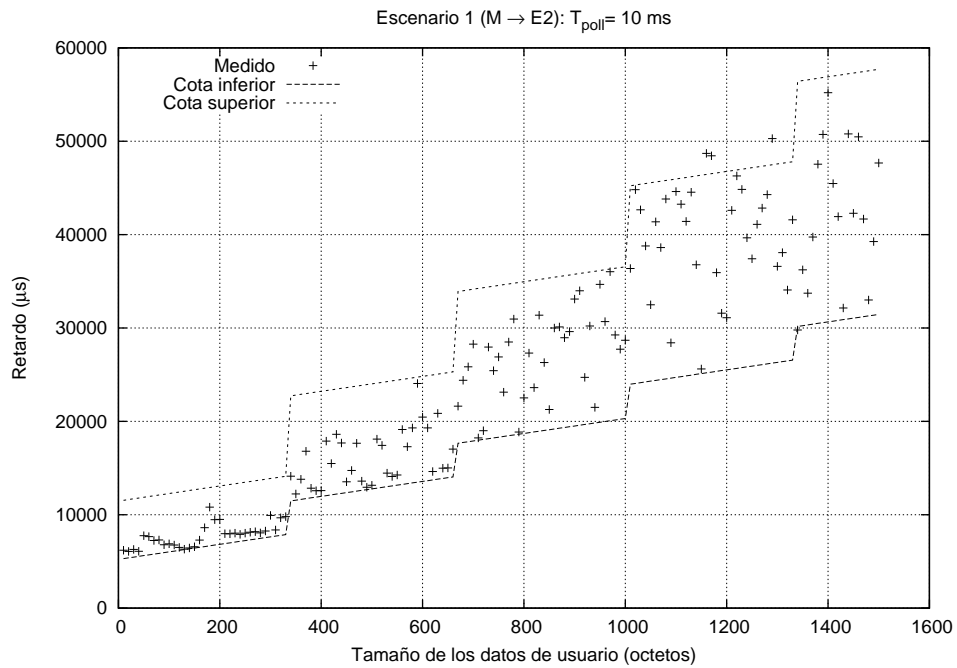
(b) Escenario 2: Transmisión del esclavo E3 al maestro

Figura 5.52: Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.20 y 5.21, y retardo medido en las transmisiones de los esclavos E1 y E2 al maestro, con paquetes DH y un valor del parámetro T_{poll} de 5 ms

5.3 Efecto de la configuración de la calidad de servicio



(a) Escenario 1: Transmisión del maestro al esclavo E1



(b) Escenario 1: Transmisión del maestro al esclavo E2

Figura 5.53: Cotas mínima y máxima, dadas por las ecuaciones 5.17 y 5.18, y retardo medido en las transmisiones del maestro a los esclavos E1 y E2, con paquetes DH y un valor del parámetro T_{poll} de 10 ms

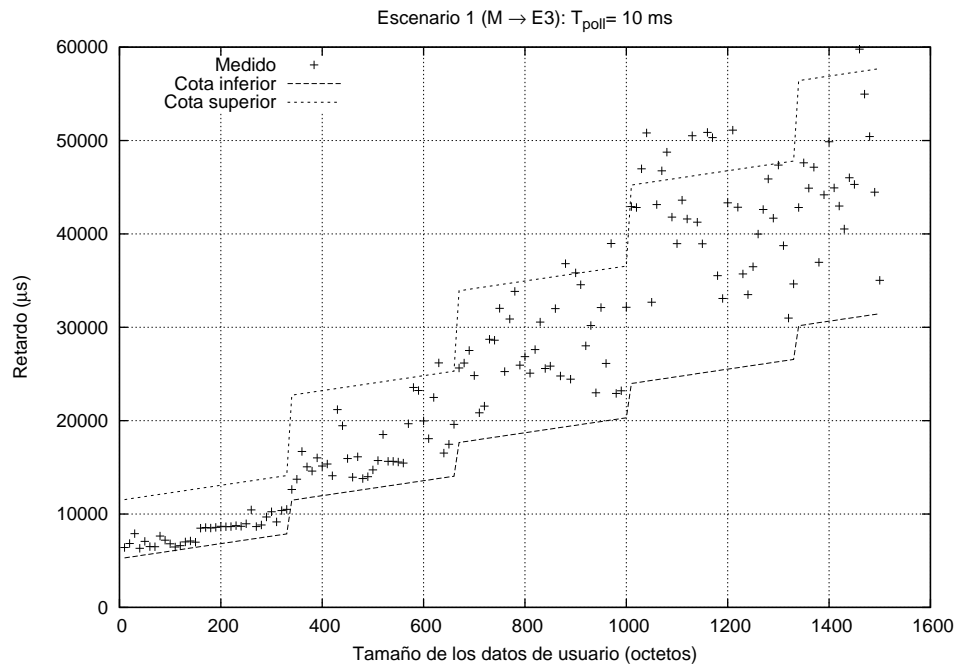


Figura 5.54: Escenario 1: Cotas mínima y máxima, dadas por las ecuaciones 5.17 y 5.18, y retardo medido en la transmisión del maestro al esclavo E3, con paquetes DH y un valor del parámetro T_{poll} de 10 ms

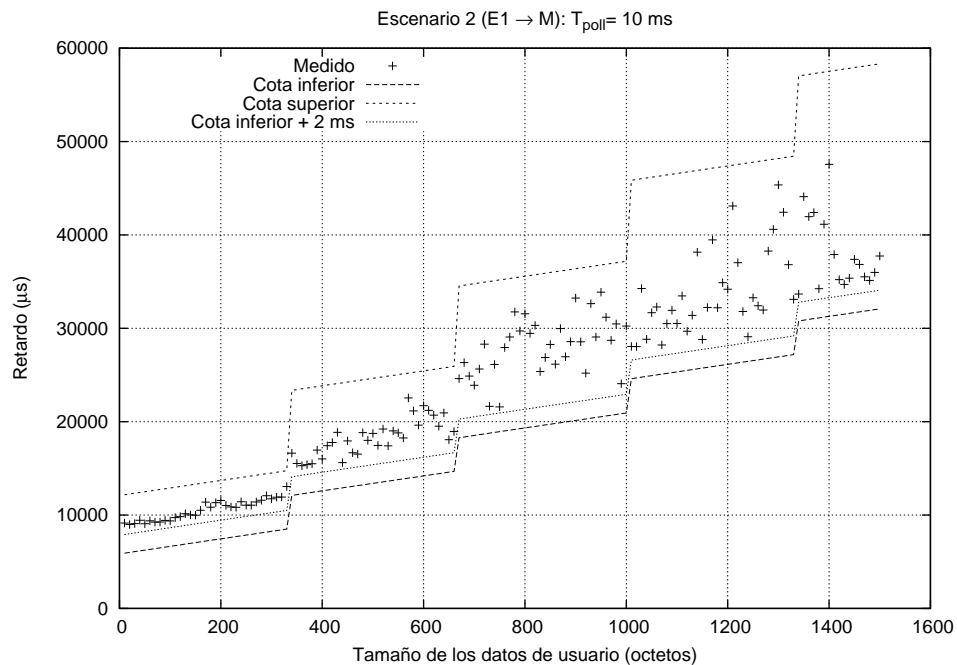
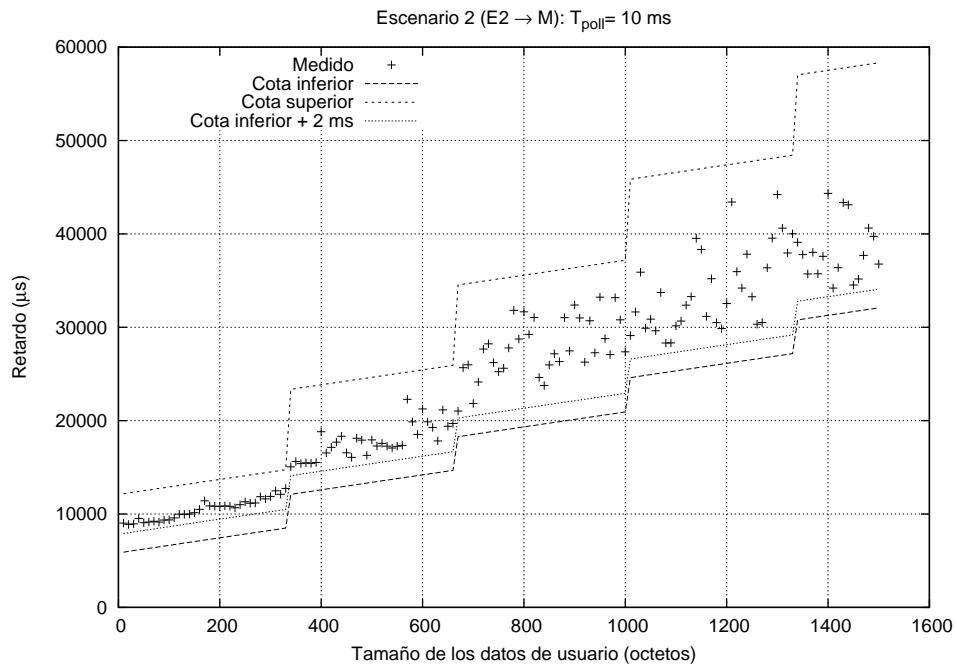
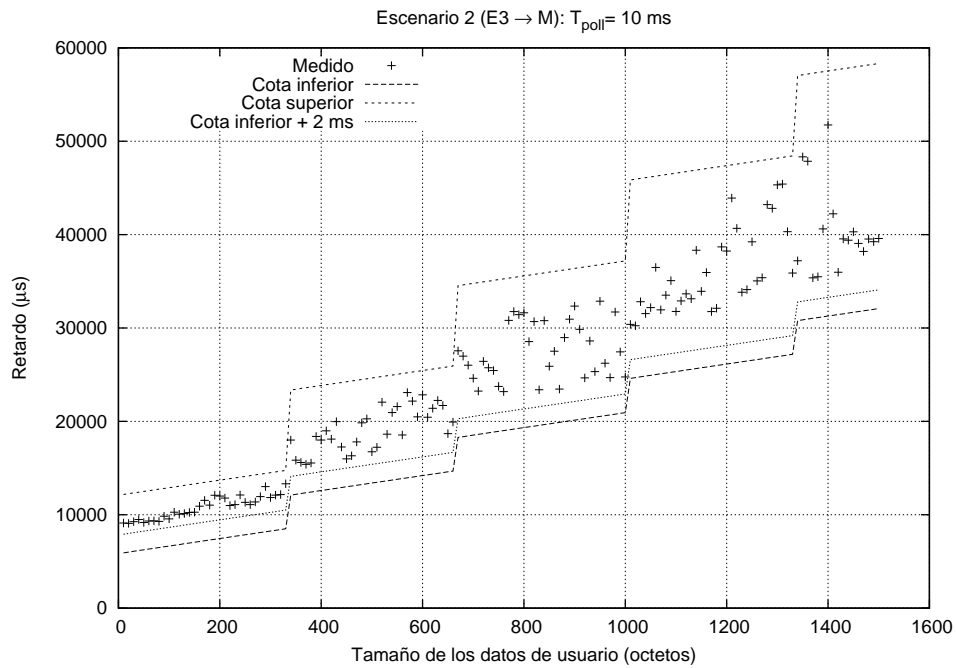


Figura 5.55: Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.20 y 5.21, y retardo medido en la transmisión del esclavo E1 al maestro, con paquetes DH y un valor del parámetro T_{poll} de 10 ms

5.3 Efecto de la configuración de la calidad de servicio

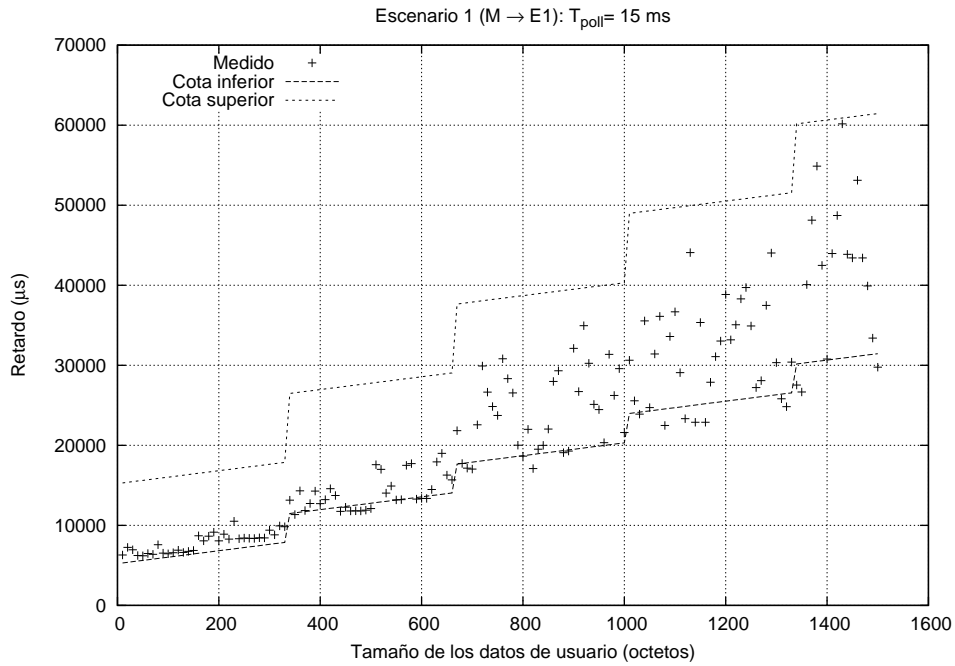


(a) Escenario 2: Transmisión del esclavo E2 al maestro

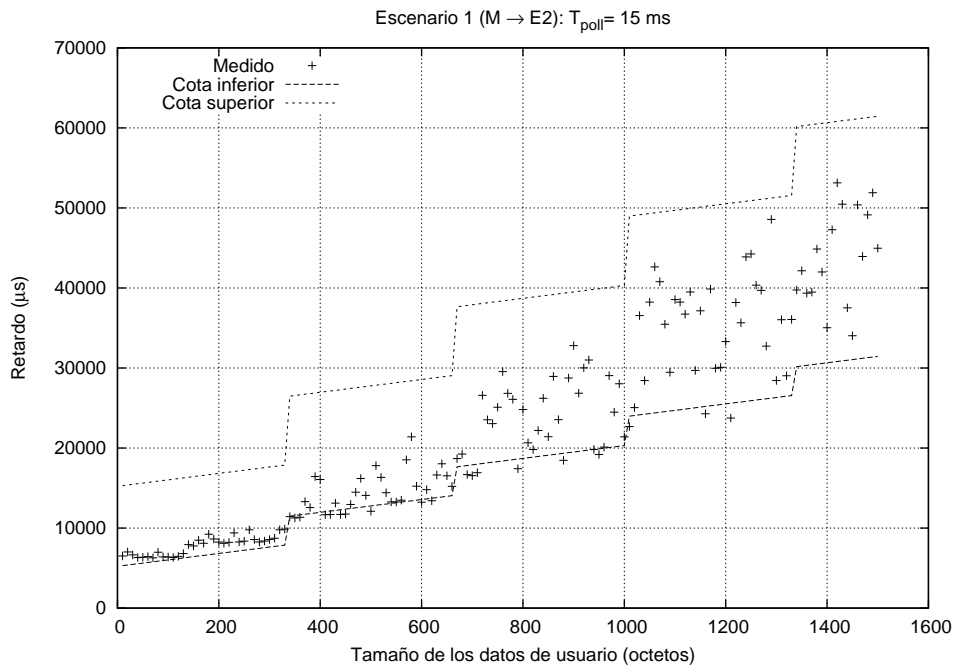


(b) Escenario 2: Transmisión del esclavo E3 al maestro

Figura 5.56: Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.20 y 5.21, y retardo medido en las transmisiones de los esclavos E1 y E2 al maestro, con paquetes DH y un valor del parámetro T_{poll} de 10 ms



(a) Escenario 1: Transmisión del maestro al esclavo E1



(b) Escenario 1: Transmisión del maestro al esclavo E2

Figura 5.57: Escenario 1: Cotas mínima y máxima, dadas por las ecuaciones 5.17 y 5.18, y retardo medido en las transmisiones del maestro a los esclavos E1 y E2, con paquetes DH y un valor del parámetro T_{poll} de 15 ms

5.3 Efecto de la configuración de la calidad de servicio

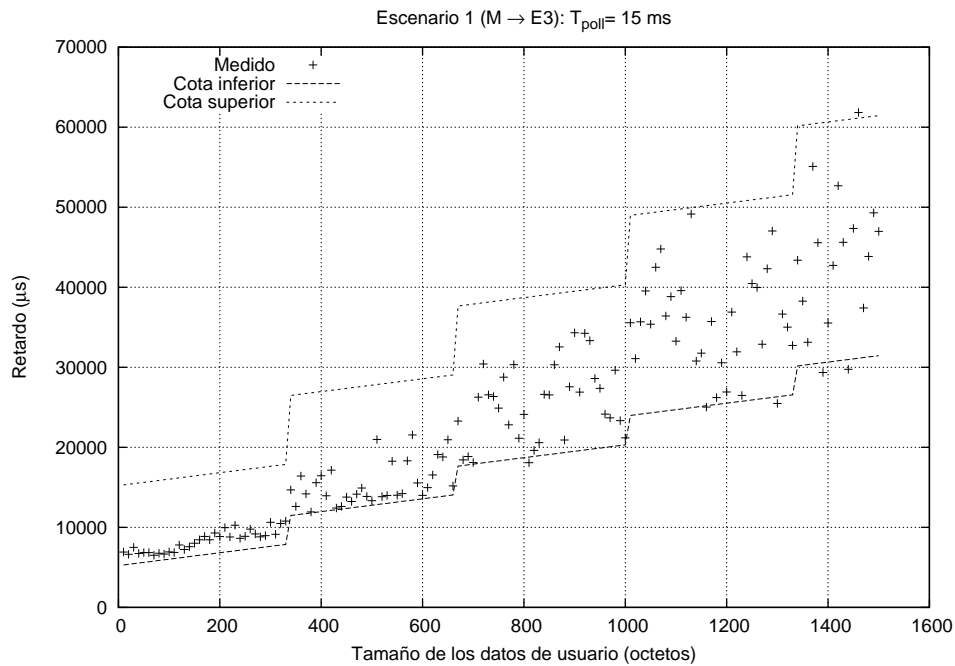


Figura 5.58: Escenario 1: Cotas mínima y máxima, dadas por las ecuaciones 5.17 y 5.18, y retardo medido en la transmisión del maestro al esclavo E3, con paquetes DH y un valor del parámetro T_{poll} de 15 ms

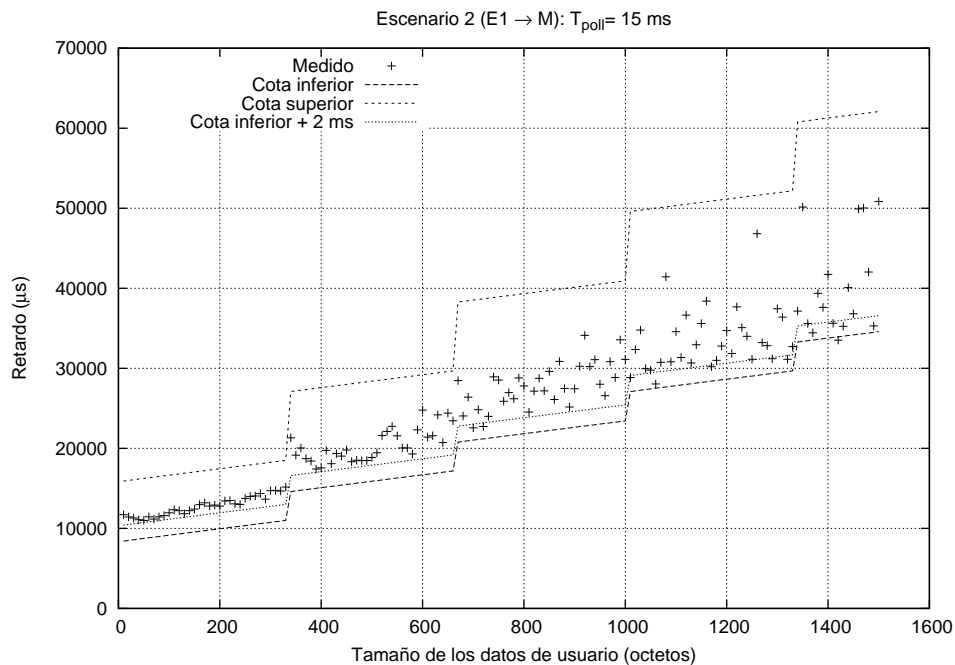
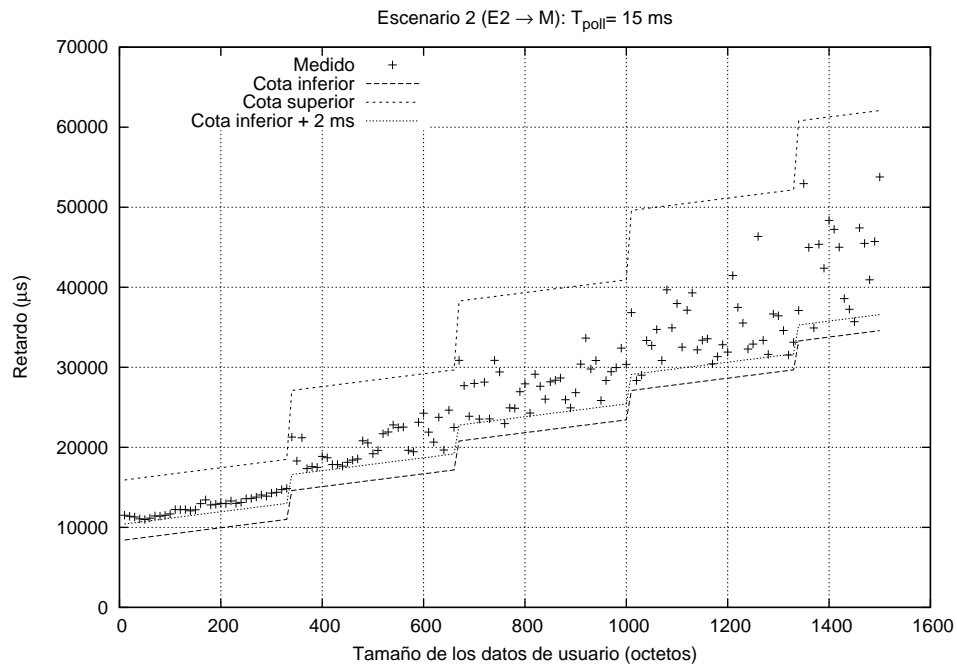
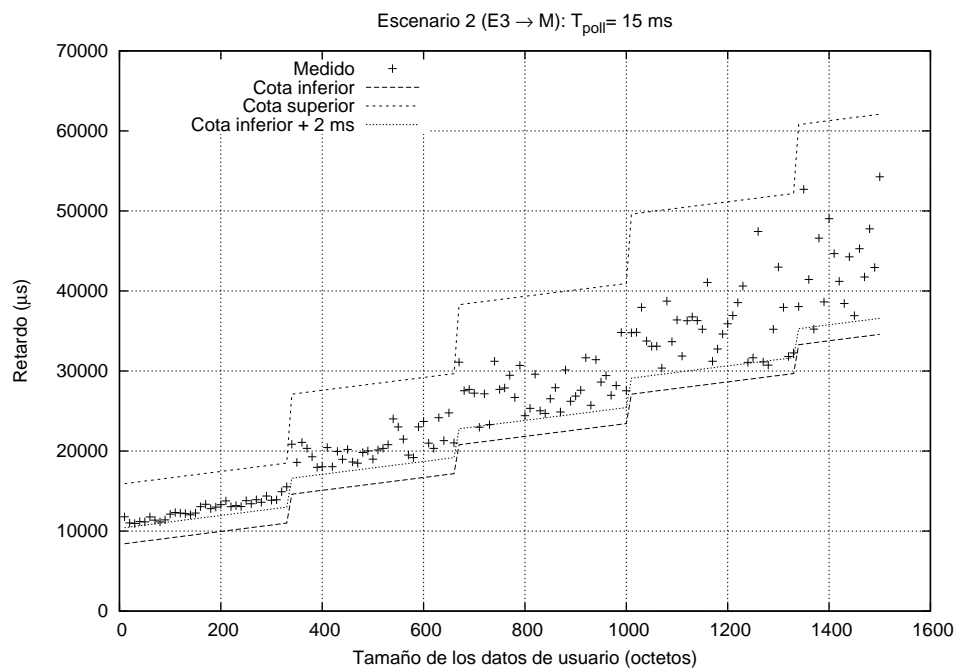


Figura 5.59: Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.20 y 5.21, y retardo medido en la transmisión del esclavo E1 al maestro, con paquetes DH y un valor del parámetro T_{poll} de 15 ms



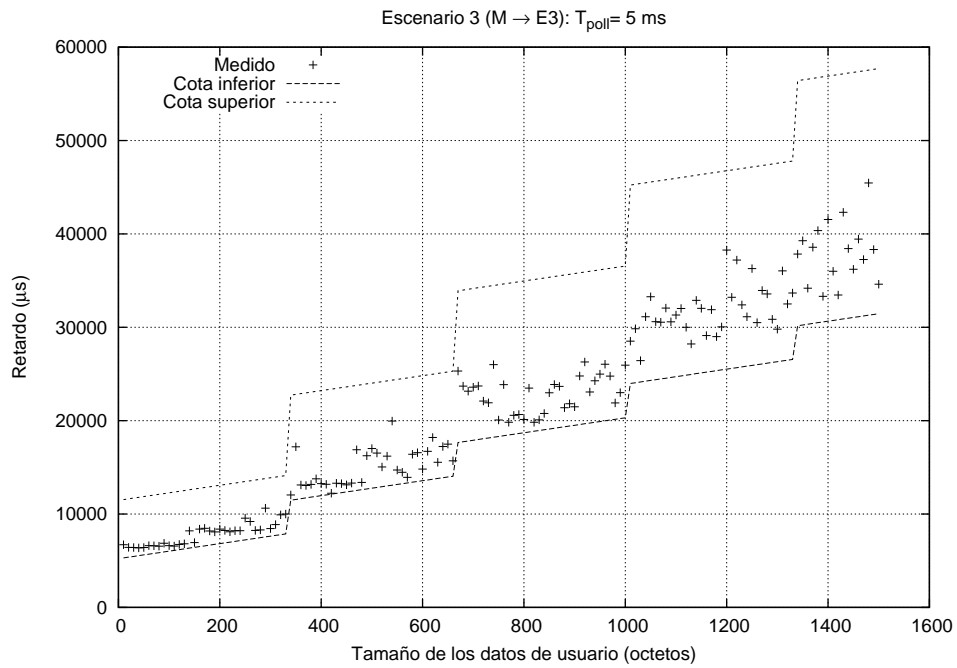
(a) Escenario 2: Transmisión del esclavo E2 al maestro



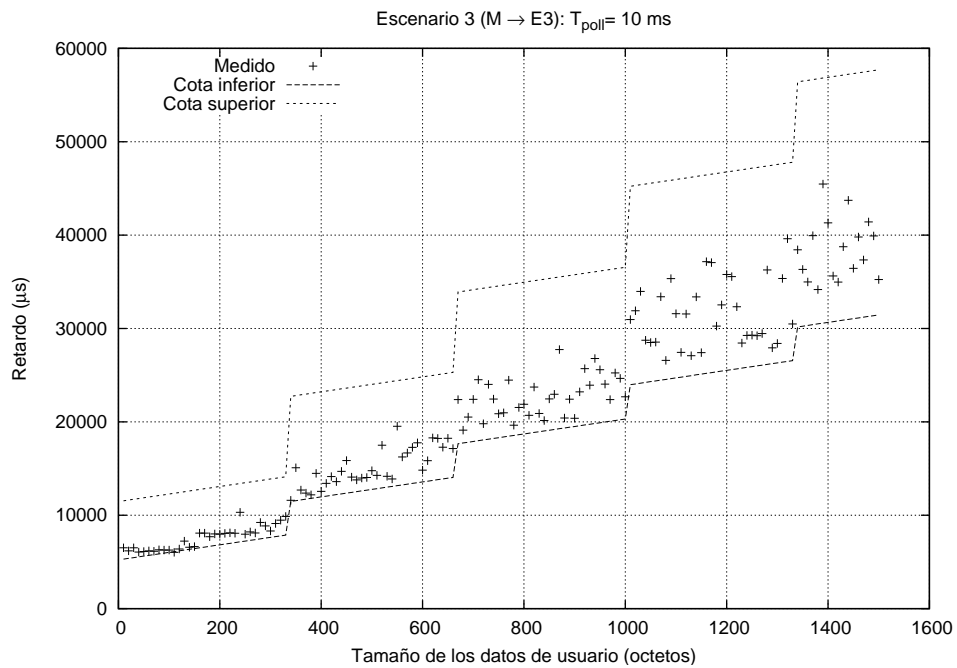
(b) Escenario 2: Transmisión del esclavo E3 al maestro

Figura 5.60: Escenario 2: Cotas mínima y máxima, dadas por las ecuaciones 5.20 y 5.21, y retardo medido en las transmisiones de los esclavos E1 y E2 al maestro, con paquetes DH y un valor del parámetro T_{poll} de 15 ms

5.3 Efecto de la configuración de la calidad de servicio



(a) Escenario 3: Transmisión del maestro al esclavo E3 con $T_{poll} = 5 \text{ ms}$



(b) Escenario 3: Transmisión del maestro al esclavo E3 con $T_{poll} = 10 \text{ ms}$

Figura 5.61: Cotas mínima y máxima, dadas por las ecuaciones 5.17 y 5.18, y retardo medido en las transmisiones del maestro a un esclavo, con paquetes DH (cuando los otros dos esclavos transmiten al maestro)

5.4. Efecto de las retransmisiones

Como ya se ha explicado anteriormente, 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. Aquí hay que tener en cuenta que, en el entorno de medida, no es factible obtener la tasa BER directamente. Por tanto, para estimarla se ha utilizado el comando *HCI_Get_Link_Quality* (véase el apartado 4.9.3 de la parte H:1 de la especificación [Bluetooth SIG01c]) 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, representada en la figura 5.62:

$$BER = \begin{cases} (255 - LQ) \cdot 0,25 \cdot 10^{-4} & 215 \leq LQ \leq 255 \\ 0,001 + (214 - LQ) \cdot 8 \cdot 10^{-4} & 90 \leq LQ \leq 215 \\ 0,1 + (89 - LQ) \cdot 64 \cdot 10^{-4} & 0 \leq LQ \leq 90 \end{cases} \quad (5.22)$$

Tomando en consideración que el fabricante indica que el cálculo de la tasa BER se realiza para todas las partes de los paquetes cuya información se protege con codificación FEC, la medida del impacto de la ocurrencia de retransmisiones se ha realizado exclusivamente para paquetes DM.

En particular, las medidas del retardo para evaluar el efecto de la ocurrencia de retransmisiones se han llevado a cabo, para el perfil SPP, en una *piconet* de dos dispositivos, en ambos sentidos de comunicación, y configurando la calidad

de servicio mediante el comando *HCI_QoS_Setup* para valores del parámetro *Latency* de 5 y 10 ms.

Para inducir pérdidas en las transmisiones, se ha retirado la caja metálica, utilizada para aislar los dispositivos Bluetooth, y se ha añadido un elemento interferente. La interferencia se ha provocado situando los dispositivos a 60 cm de un punto de acceso 802.11 y transmitiendo tráfico UDP, con una tasa de 10 Mb/s, entre el equipo al que se conectan los módulos Bluetooth y otro PC perteneciente a la WLAN, según se ilustra en la figura 5.63. No obstante, es conveniente aclarar que las medidas no se han podido ejecutar en un entorno controlado, y por tanto, no se puede garantizar que fuera ésta la única fuente interferente.

Las pruebas, como en los casos anteriores, han contemplado sistemáticamente un rango amplio de tamaños de los datos a transmitir: de 10 a 1500 octetos, con incrementos de 10 octetos. El procedimiento de medida ha consistido en ejecutar un servidor y un cliente RFCOMM, de forma que, al igual que en los casos anteriores, será el servidor el que realice las 1000 transmisiones para cada longitud del intervalo, si es el sentido esclavo-maestro el que se desea caracterizar, y el cliente, en caso contrario. Sin embargo, en relación con otros escenarios ha sido necesario incrementar el periodo de transmisión entre envíos consecutivos a $T = 200\text{ ms}$. La razón que justifica esta decisión es evitar que la aparición del efecto de encolamiento en los *buffers* del dispositivo (provocado por las retransmisiones) distorsione la medida realizada.

El comando *HCI_Get_Link_Quality* se envía a los módulos Bluetooth al finalizar las 1000 repeticiones que se realizan para cada tamaño y el valor del

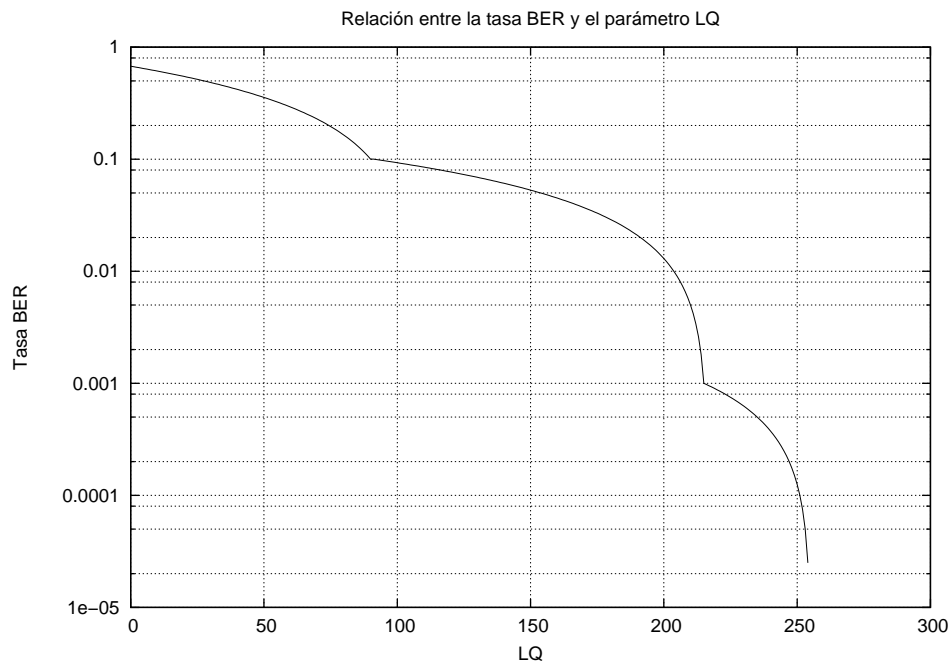


Figura 5.62: Representación de la ecuación 5.22, que define la relación entre la tasa BER y el parámetro LQ, según especifica el fabricante CSR [CSR]

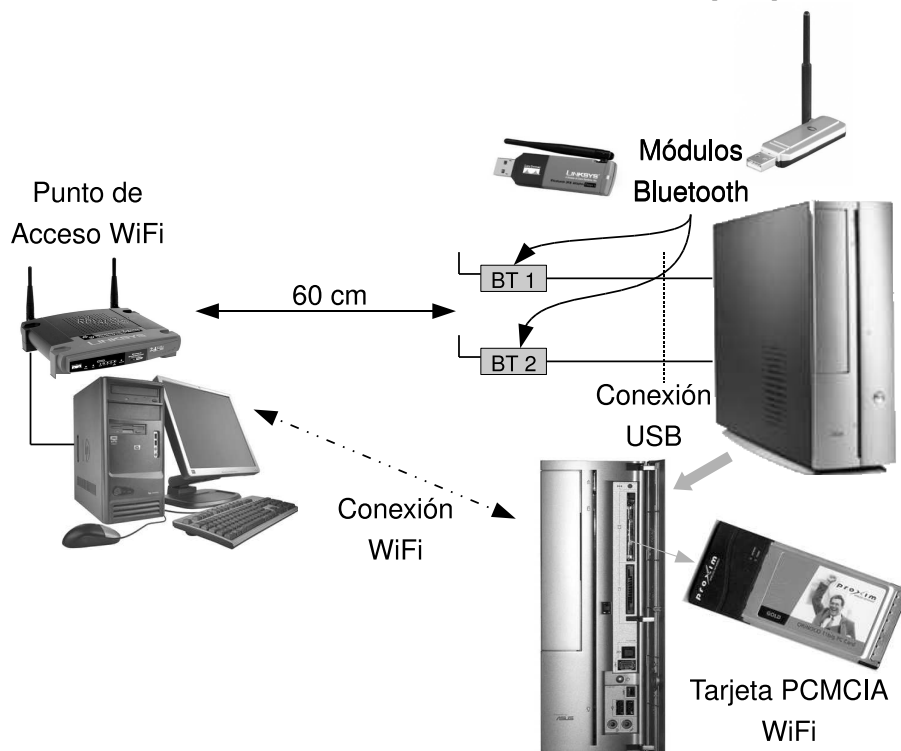


Figura 5.63: Esquema de pruebas utilizado en la evaluación del impacto de las retransmisiones sobre el retardo en transmisiones Bluetooth, utilizando el perfil SPP

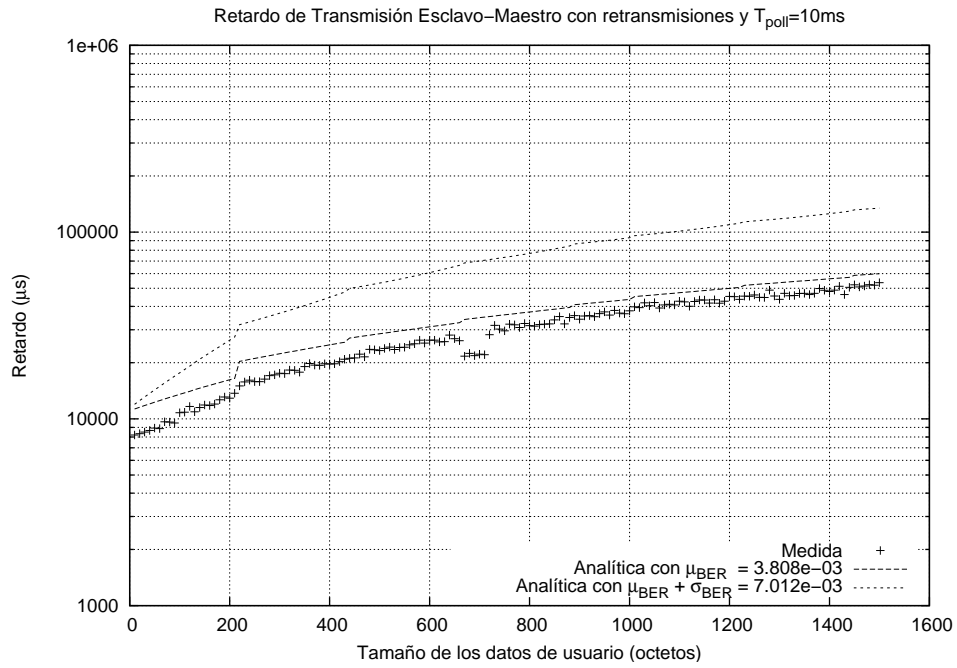
parámetro LQ retornado se almacena en un fichero, junto a la longitud para la que ha sido obtenido.

En las figuras 5.64–5.69 se incluyen los resultados. Concretamente, las subfiguras (a) muestran el retardo medio medido y el obtenido analíticamente, a partir de las ecuaciones 4.43 y 4.44, incluyendo las correcciones indicadas en los apartados 5.3.1.5 y 5.3.1.4, y teniendo en cuenta que, para que las ecuaciones 4.43 y 4.44 sean cotas máximas, se ha de particularizar $t_{RTx}(N)$ al valor máximo, $t_{RTx_{max}}(N)$.

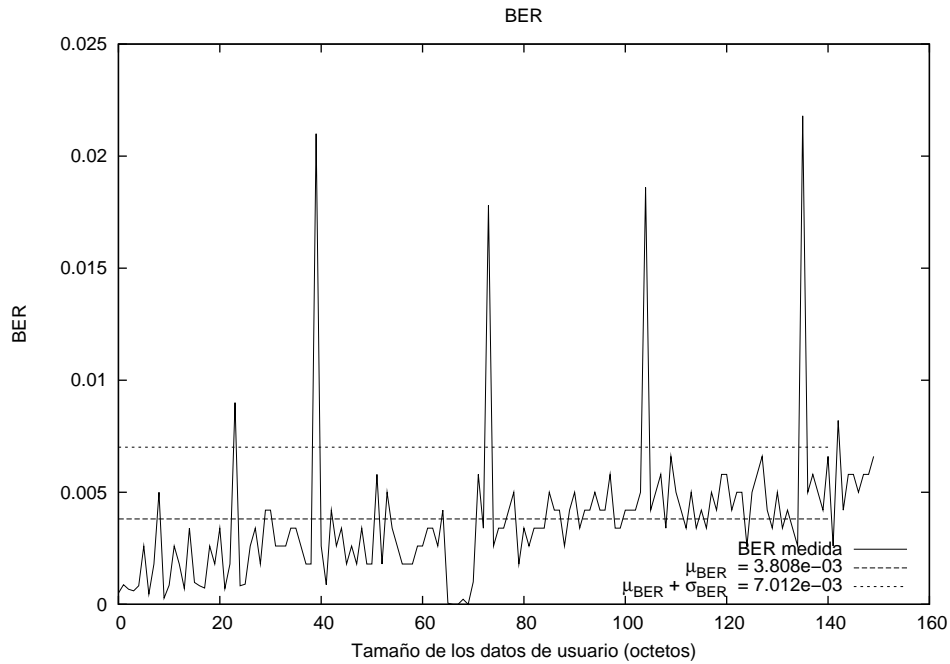
Adicionalmente, conviene precisar que la cota de retardo no ha sido calculada para la tasa BER estimada puntualmente para cada longitud, sino para el promedio de las tasas BER obtenidas para el rango completo de longitudes, notada en todas las gráficas como μ_{BER} . Además, teniendo en cuenta que el fabricante advierte de la falta de precisión en la estimación de la calidad del enlace proporcionada por el dispositivo, también se ha representado la cota del retardo correspondiente a la tasa BER media más la desviación típica, σ_{BER} , es decir, $\mu_{BER} + \sigma_{BER}$.

En las figuras 5.64–5.69, las subfiguras (b) representan: la tasa BER estimada para cada tamaño de los datos de usuario, el valor promedio, μ_{BER} , y la suma $\mu_{BER} + \sigma_{BER}$. La finalidad es complementar cada una de las gráficas de retardo, ilustrando que, a longitudes inferiores, el valor de la tasa BER estimada difiere apreciablemente de la media μ_{BER} , y justificando así que el retardo calculado analíticamente a partir de este valor (μ_{BER}) proporcione una predicción más deficiente que la resultante a tamaños superiores.

Todas las gráficas mostradas, a excepción de la incluida en la figura 5.67, ponen de manifiesto que el retardo calculado teóricamente para la tasa BER media, μ_{BER} , proporciona verdaderamente una muy buena aproximación que, además, para algunos casos, como los representados en las figuras 5.65 y 5.68, tiene la particularidad de que se ajusta fielmente al comportamiento obtenido empíricamente, sobre todo, para longitudes superiores a los 200 octetos. Por contra, en la figura 5.67 la cota superior del retardo que no es excedida por el retardo medio medido, es la obtenida para $\mu_{BER} + \sigma_{BER}$. No obstante, aún así, en este caso hay que tener en cuenta que la desviación estándar es muy pequeña, por lo que no se aleja mucho del límite teórico calculado para la tasa BER promediada.

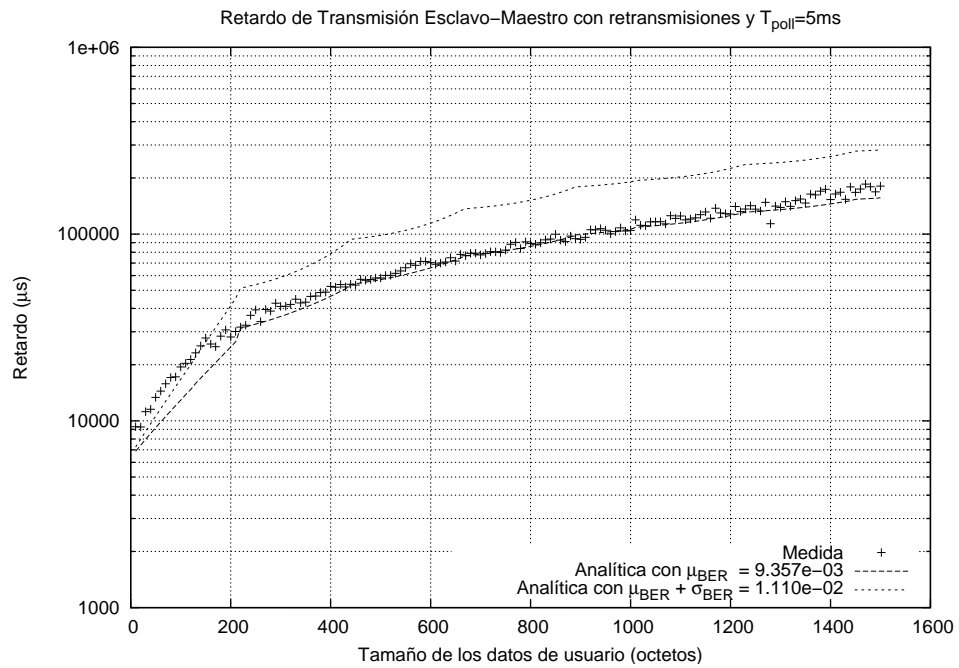


(a) Retardo de transmisión: medido y teórico

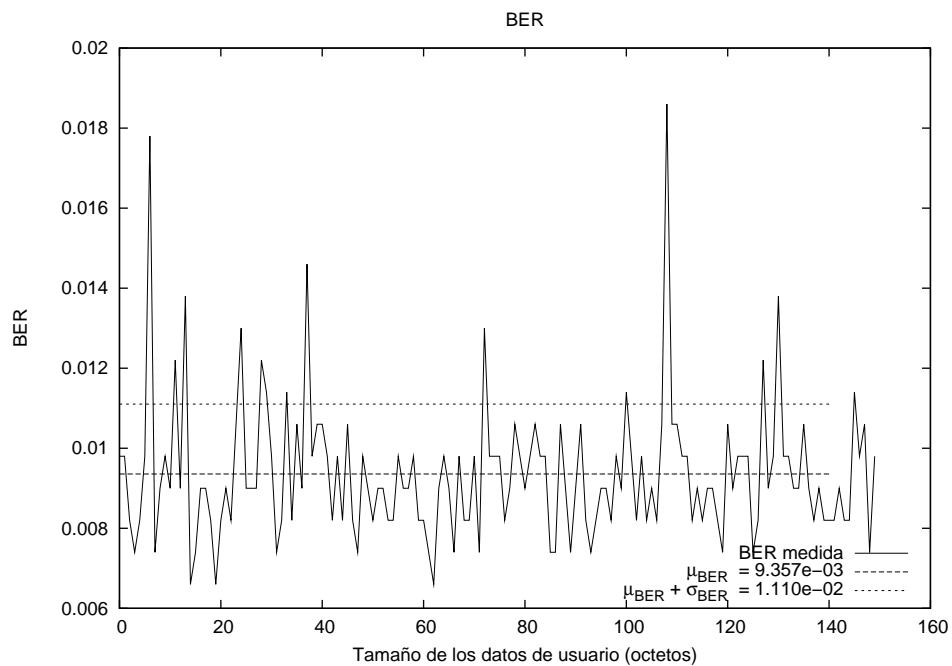


(b) Tasa BER

Figura 5.64: Transmisión esclavo-maestro con pérdidas para un valor del parámetro T_{poll} de 10 ms. BER media estimada: $3,808 \times 10^{-3}$

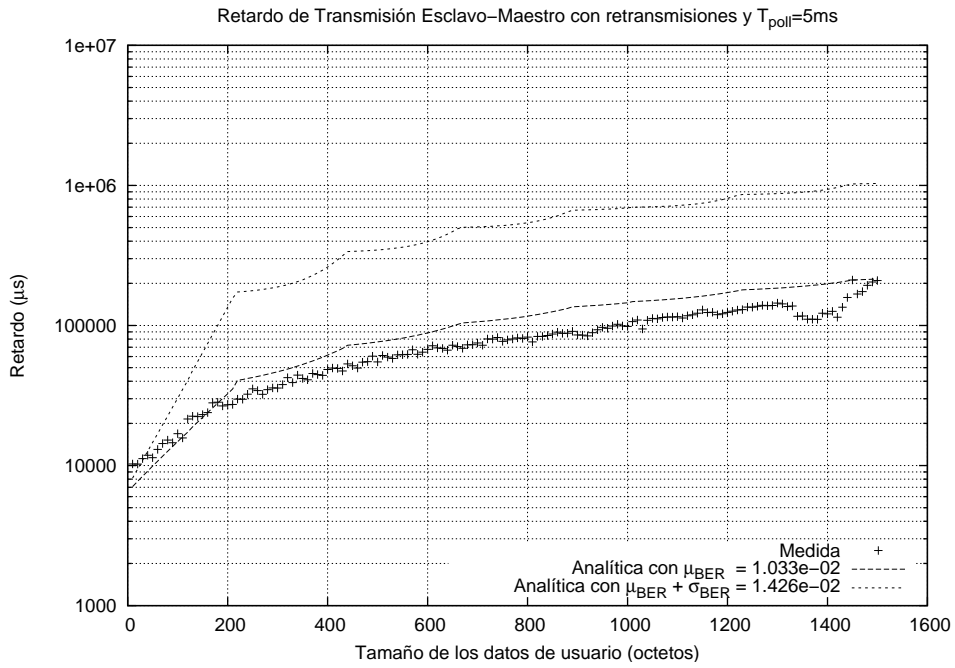


(a) Retardo de transmisión: medido y teórico

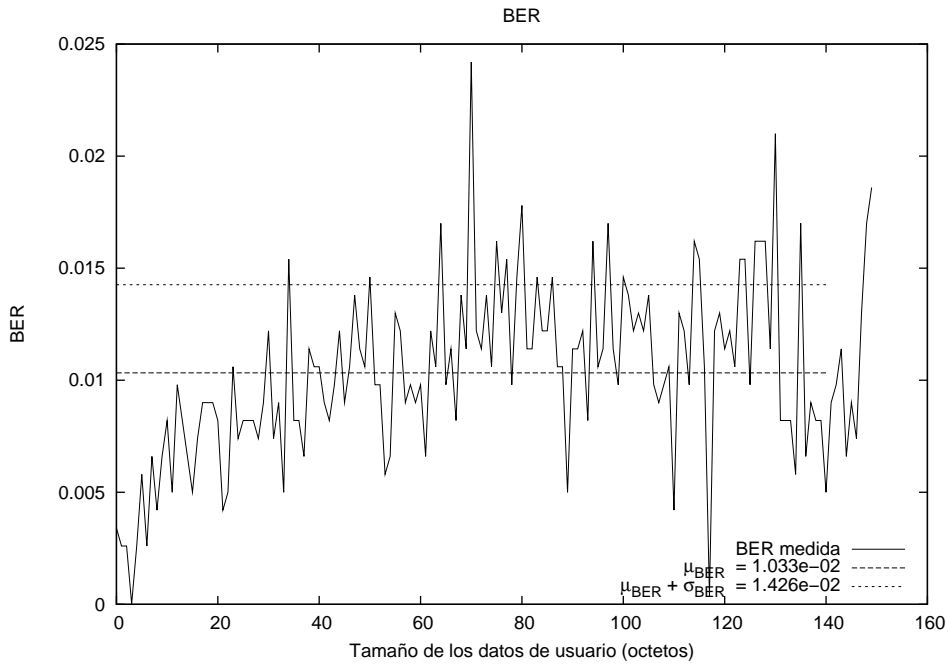


(b) Tasa BER

Figura 5.65: Transmisión esclavo-maestro con pérdidas para un valor del parámetro T_{poll} de 5 ms. BER media estimada: $9,357 \times 10^{-3}$

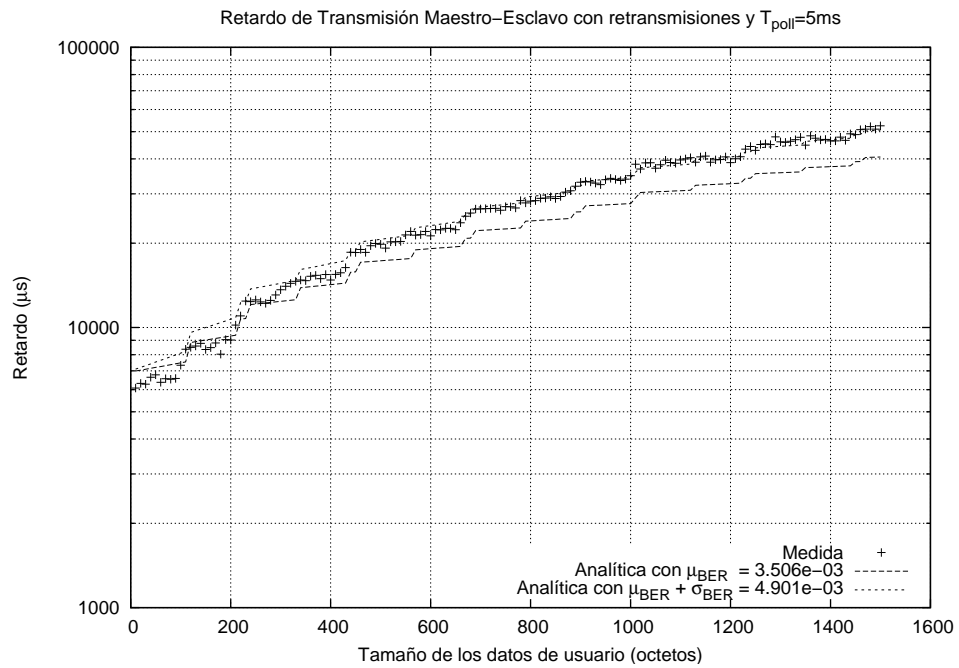


(a) Retardo de transmisión: medido y teórico

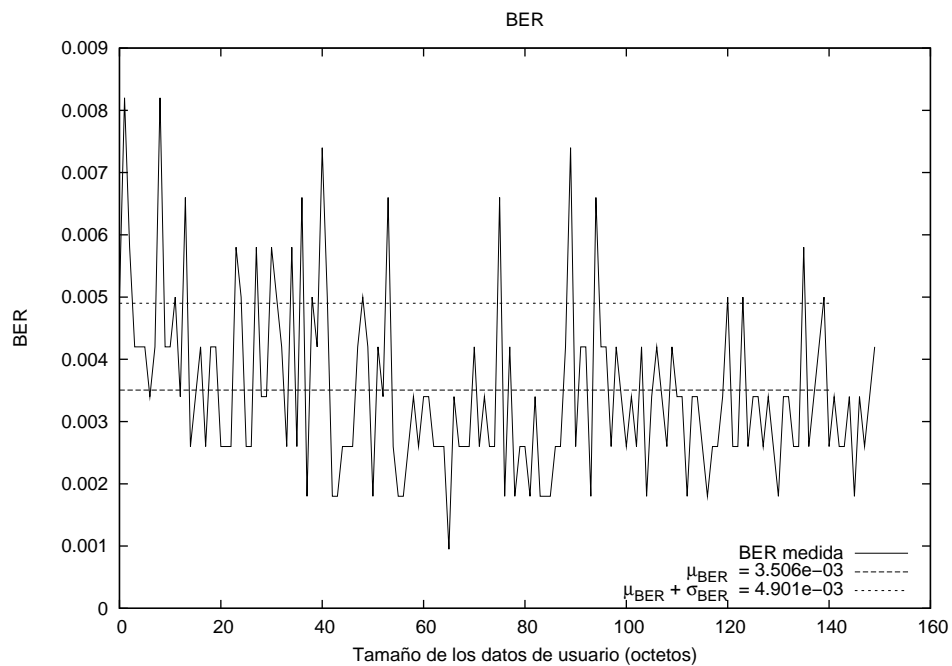


(b) Tasa BER

Figura 5.66: Transmisión esclavo-maestro con pérdidas para un valor del parámetro T_{poll} de 5 ms. BER media estimada: $1,033 \times 10^{-2}$

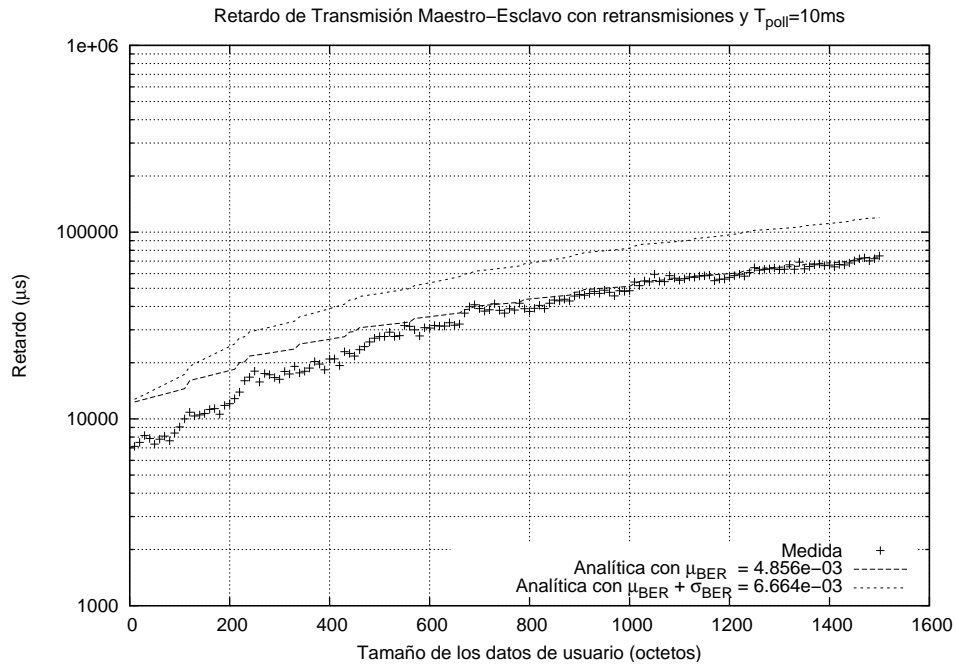


(a) Retardo de transmisión: medido y teórico

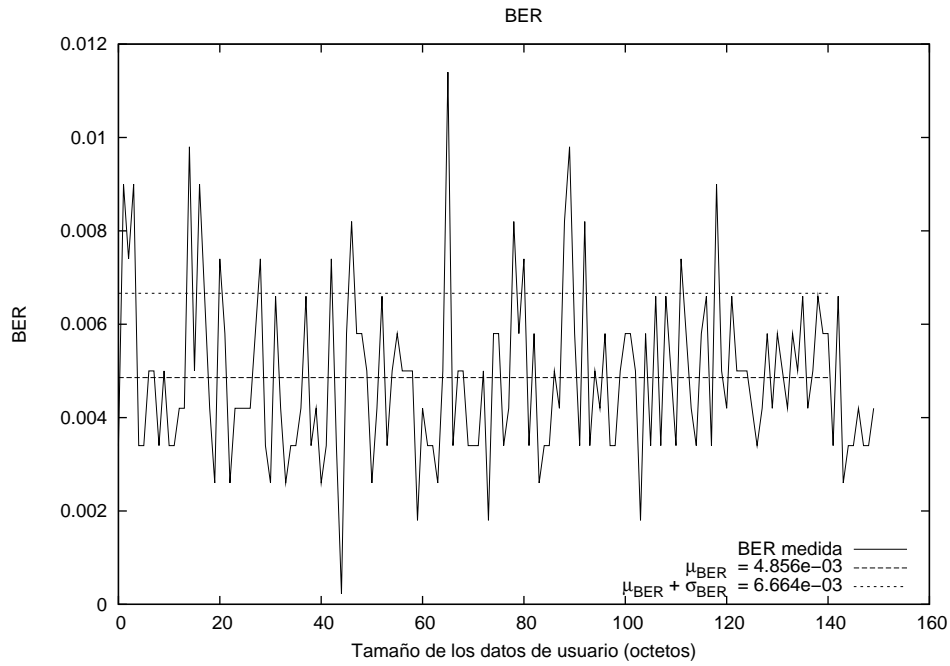


(b) Tasa BER

Figura 5.67: Transmisión maestro-esclavo con pérdidas para un valor del parámetro T_{poll} de 5 ms. BER media estimada: $3,506 \times 10^{-3}$

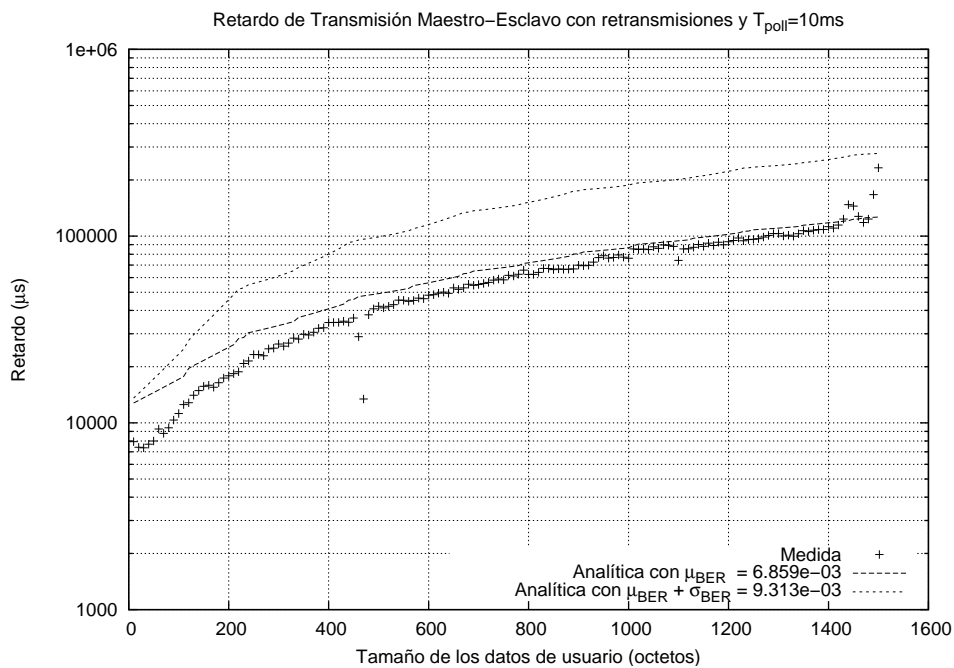


(a) Retardo de transmisión: medido y teórico

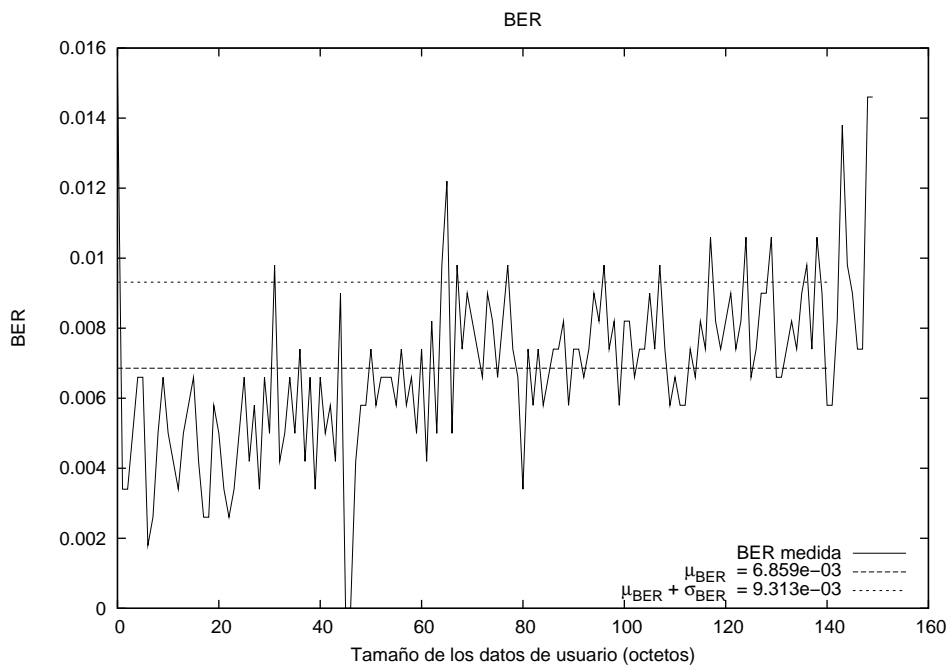


(b) Tasa BER

Figura 5.68: Transmisión maestro-esclavo con retransmisiones para un valor del parámetro T_{poll} de 10 ms. BER media estimada: $4,856 \times 10^{-3}$



(a) Retardo de transmisión: medido y teórico



(b) Tasa BER

Figura 5.69: Transmisión maestro-esclavo con pérdidas para un valor del parámetro T_{poll} de 10 ms. BER media estimada = $6,85 \times 10^{-3}$

Capítulo 6

Conclusiones y líneas futuras

En el diseño de aplicaciones para redes de área personal inalámbricas los factores claves que deben tenerse en cuenta son, fundamentalmente, la interoperatividad y el soporte de calidad de servicio, requisito imprescindible para garantizar la utilización del ancho de banda de las aplicaciones móviles basadas en este tipo de redes.

Esta tesis se ha centrado en el estudio del comportamiento de Bluetooth que, hoy por hoy, es la tecnología más extendida para la formación de redes de área personal y que, gracias a que su especificación incluye la definición de perfiles, garantiza la interoperatividad.

Sin embargo, hasta el momento, en el ámbito de las redes WPAN Bluetooth, un número significativo de trabajos no contemplan el uso de perfiles, considerando que los datos se entregan directamente a niveles inferiores de la arquitectura Bluetooth, obviando así el *overhead* introducido por las capas

superiores. Al respecto, es conveniente aclarar que esta opción no es factible en sistemas en los que se integran dispositivos reales con un determinado perfil, normalmente SPP, como es el caso de los prototipos de sistemas de telemedicina sobre los que se ha trabajado en los últimos años y que se han presentado en [Moron05a, Moron05b, Moron05c, Moron07a, Moron07c].

Es precisamente el hecho de que los sensores médicos utilizados en los prototipos de sistemas de monitorización descritos en estos trabajos implementen el perfil SPP lo que ha motivado, en esta tesis, el estudio de rendimiento de Bluetooth, en términos de retardo extremo a extremo, para este perfil.

También se ha querido comparar el comportamiento del perfil SPP con el del perfil PAN, al que se le augura una futura expansión debido a que permite el encapsulamiento de tráfico IP sobre Bluetooth, aunque actualmente no esté soportado por ningún dispositivo comercial.

Concretamente, el objetivo de esta tesis ha sido estudiar el rendimiento de una WPAN Bluetooth, en términos del retardo extremo a extremo y del *throughput*, sin prescindir del uso de perfiles y evaluando el efecto de la configuración de la calidad de servicio y de las retransmisiones.

El estudio desarrollado ha consistido en formular expresiones matemáticas para calcular el retardo de transmisión mínimo en el sentido de transmisión maestro-esclavo, dependiendo del tipo de paquete utilizado, y para un tamaño de datos de usuario arbitrario. Posteriormente, el modelo propuesto ha sido extendido para contemplar el *overhead* introducido por los niveles superiores, tanto en el caso del perfil SPP, como en el del perfil PAN. Adicionalmente,

para el perfil SPP se ha completado el estudio analítico a fin de considerar la configuración de calidad, así como la ocurrencia de retransmisiones.

En una segunda fase, se ha llevado a cabo la validación del modelo mediante caracterización empírica, empleando dispositivos Bluetooth 1.1. A raíz de los resultados obtenidos se han introducido algunas correcciones que permiten afinar el modelo, ajustándolo mejor al comportamiento observado experimentalmente.

También se ha extendido el estudio de evaluación de la configuración de calidad al caso de una *piconet* Bluetooth de dos esclavos, proporcionando cotas mínima y máxima del retardo para cada sentido de transmisión, cotas, que tras verificarse empíricamente, han sido generalizadas y de nuevo, validadas experimentalmente para una *piconet* Bluetooth de tres esclavos.

Finalmente, se han repetido las medidas realizadas para una *piconet* de un único esclavo utilizando dispositivos Bluetooth 1.2 y Bluetooth 2.0, pudiéndose concluir que las ecuaciones de retardo propuestas modelan aún mejor el comportamiento observado que en el caso de Bluetooth 1.1.

Las conclusiones extraídas del estudio realizado en esta tesis se exponen en la sección 6.1, y las líneas futuras que se proponen para continuar ampliando el trabajo desarrollado en esta tesis se esbozan, finalmente, en la sección 6.2.

6.1. Conclusiones

En los siguientes apartados se sintetizan las conclusiones correspondientes a los aspectos que a continuación se enumeran:

- Estudio y comparación del rendimiento de los perfiles PAN y SPP (apartado 6.1.1).
- Estudio del efecto de la configuración de calidad en el caso del perfil SPP (apartado 6.1.2).
- Estudio, para SPP, del efecto de las pérdidas debido a la interferencia (apartado 6.1.3).

6.1.1. Conclusiones sobre el estudio comparativo de los perfiles SPP y PAN

El objetivo en este caso ha sido el estudio del efecto del *overhead* y del *segmentation mismatch* sobre el *throughput* y el retardo extremo a extremo para el perfil SPP, implementado por una amplia variedad de dispositivos comerciales y también, para el perfil PAN, muy adecuado para ser utilizado en redes de área personal.

Las conclusiones que se desprenden del estudio llevado a cabo en relación con los perfiles SPP y PAN se pueden agrupar en dos categorías:

1. Efecto del *overhead* y del *segmentation mismatch*:

- *Perfil PAN*: La especificación de BNEP [Bluetooth SIG03b] establece el valor mínimo de la MTU de L2CAP y, en función de ésta y del máximo tamaño de la cabecera de BNEP, limita la longitud de la trama que el nivel superior podrá entregar a BNEP, garantizando así que no se produce el efecto del *segmentation mismatch*. Además, dado que los

1691 octetos a los que se fija la MTU M_B , sumados a los 4 octetos de *overhead* añadidos por L2CAP, pueden transmitirse en 3 paquetes DH5, se consigue maximizar el rendimiento, al menos, cuando se utilizan paquetes DH.

- *Perfil SPP*: Dado que la especificación de RFCOMM no establece criterios similares a los definidos para el protocolo BNEP, puede producirse una pérdida de rendimiento a causa del *segmentation mismatch*. Del estudio realizado se concluye que para minimizar este efecto, y por tanto, mejorar el *throughput*, es suficiente con cumplir dos requisitos:
 - Negociar para el parámetro $N1$ un valor igual o superior a la máxima longitud de datos a transmitir, ya que $N1$ determina el máximo número de octetos de información de usuario que es posible transportar en tramas UIH de RFCOMM.
 - Seleccionar un valor para la MTU de L2CAP igual al valor del parámetro $N1$ más los 5 octetos de sobrecarga de protocolo RFCOMM.

En definitiva, si el tamaño de MTU utilizado en cada nivel se fija teniendo en consideración el del nivel subyacente y la máxima capacidad de los paquetes que se utilizan en el nivel físico, como se hace en el caso del perfil PAN, el rendimiento no se ve perjudicado.

2. Modelo teórico:

- a) El retardo de transmisión no depende del número de *slots* ocupados por el paquete utilizado sino del número de bits enviados, de forma que la duración del paquete y el tiempo de confirmación sólo afectan

al retardo cuando la longitud de los datos a transmitir supera el máximo tamaño del paquete de mayor capacidad. Gráficamente, este comportamiento se traduce en:

- 1) Un crecimiento lineal del retardo, siempre que no se exceda la capacidad de un paquete de 5 *slots*.
 - 2) Un salto incremental de 6 *slots*, que sucede cada vez que se produce el llenado de un paquete DH5 o DM5, y que ha de contemplarse debido a que antes de proseguir enviando el siguiente fragmento, hay que esperar a que el extremo remoto confirme el último paquete transmitido.
- b) Análogamente, el cálculo del *throughput* real ha de contemplar que la transmisión de datos de usuario de longitudes elevadas requiere el envío de más de un paquete del nivel BB, teniendo en cuenta que el esquema de transmisión utilizado por Bluetooth es TDD, y considerando el tiempo de guarda existente en el último (o único) *slot* ocupado por un paquete.
- c) El retardo previsto teóricamente y el medido experimentalmente, tanto para el perfil SPP como para el perfil PAN, difieren en un *offset*, que como se ha comprobado al evaluar el impacto de la configuración de calidad, está relacionado con el intervalo de *poll* empleado.

6.1.2. Conclusiones sobre el estudio del efecto de la configuración de calidad realizado para el perfil SPP

El estudio del efecto de la configuración de calidad realizado para el perfil SPP se ha llevado en las siguientes fases:

1. **Elaboración de un modelo formal:** Se han propuesto expresiones analíticas que proporcionan una cota máxima del retardo de transmisión en una *piconet* de dos dispositivos y en cada uno de los sentidos de comunicación: maestro-esclavo y esclavo-maestro. Las ecuaciones formuladas se fundamentan en el hecho de que en las especificaciones de Bluetooth (parte C de la especificación [Bluetooth SIG01c]) se indica que una unidad que esté actuando como maestro sólo está obligada a comunicarse con cada esclavo de la *piconet* una vez cada n_{poll} *slots*. La consecuencia directa de este criterio conlleva el hecho de que los fragmentos consecutivos que se generan cuando el tamaño de los datos a transmitir excede la capacidad máxima de un paquete de 5 *slots* podrían estar separados hasta un máximo de n_{poll} *slots*.
2. **Caracterización empírica:** Para Bluetooth 1.1 se han realizado medidas de los retardos de transmisión en cada uno de los sentidos de comunicación y para distintos valores del intervalo de *poll*, abarcando un rango de longitudes comprendido entre 10 y 1500 octetos, de forma que el retardo obtenido para cada tamaño ha sido el promedio de los resultados de 1000 repeticiones.

A partir del comportamiento observado experimentalmente, se extraen las siguientes conclusiones:

a) En los sentidos de transmisión maestro-esclavo y esclavo-maestro:

- 1) Los saltos equiespaciados que se producen en la gráfica del retardo cada vez que se completa un paquete de 5 *slots* no presentan dependencia del valor del intervalo de *poll* configurado. Esto es, los fragmentos consecutivos que se generan cuando el tamaño de los datos a transmitir excede la capacidad máxima de un paquete de 5 *slots* no están separados n_{poll} *slots*, como se había previsto teóricamente, sino 6 *slots*.
- 2) Las curvas de retardo obtenidas para los distintos valores del intervalo de *poll* ponen de manifiesto que existe un *offset* de $T_{poll}/2$, que es la media de una variable aleatoria que sigue una distribución uniforme entre 0 y T_{poll} , y que se corresponde con el tiempo medio transcurrido desde que el *Host* solicita el envío de datos, hasta que el nivel BB transmite esos datos.

b) En el sentido de transmisión maestro-esclavo:

- 1) La configuración de valores de T_{poll} por debajo de 10 ms no conduce a la obtención de un retardo menor.
- 2) Se ha comprobado que, por cada salto predicho teóricamente, se producen dos escalones. Este comportamiento se atribuye a que el esclavo no entrega los datos, procedentes del maestro, al *Host* (PC) hasta que no transcurre, al menos, el tiempo correspondiente a un paquete de 1, 3 ó 5 *slots*, completo.

c) En el sentido de transmisión esclavo-maestro:

- 1) El valor mínimo de T_{poll} que puede configurarse de forma que se traduzca en un retardo menor que el obtenido para otros

intervalos de *poll* es de 5 ms.

- 2) Las curvas de retardo obtenidas para los distintos valores del intervalo de *poll* revelan que además de un *offset* de $T_{poll}/2$, existe un incremento de 2 ms.
- 3) El maestro no confirma el paquete de 5 *slots* que transporta un fragmento inicial, y no único, en el *slot* inmediatamente posterior a la recepción.

3. **Corrección y evaluación del modelo teórico:** El modelo teórico ha sido modificado incorporando los ajustes correspondientes a las observaciones anteriores. Además, se ha constatado que los resultados experimentales responden al comportamiento teórico.
4. **Extensión del modelo para *piconets* de más de dos dispositivos:** Esta parte del estudio ha sido abordada comenzando por proponer unas cotas mínimas y máximas para el retardo de transmisión en una *piconet* con tres dispositivos, considerando que, en presencia de un segundo esclavo, el maestro va a direccionar a los dos esclavos, alternadamente, pudiendo ocurrir que las transmisiones de datos se sincronicen en el tiempo o no. Las cotas formuladas analíticamente se han verificado de modo empírico y, posteriormente, han sido generalizadas para una *piconet* de n_e esclavos, comprobando que se satisfacen al agregar un cuarto dispositivo, es decir, para el caso particular de una *piconet* de tres esclavos ($n_e = 3$).
5. **Validación del modelo para Bluetooth 1.2 y 2.0:** Las medidas realizadas para las siguientes versiones de Bluetooth, que se han incluido en el apéndice A, han permitido constatar no sólo la vigencia del modelo, sino extraer las siguientes conclusiones, que contribuyen a reforzar su validez:

- a) Desaparición del *offset* de 2 ms presente en las transmisiones esclavo-maestro.
- b) Independencia del retardo de transmisión en el sentido maestro-esclavo respecto del intervalo de *poll* configurado.
- c) Mejora en el comportamiento del esclavo en modo recepción, ya que no está presente el doble escalón, cuya existencia se había imputado al hecho de que el nivel BB del esclavo no entregaba los datos, procedentes del maestro, al *Host* (PC) hasta que no transcurría, al menos, el tiempo correspondiente a un paquete de 1, 3 ó 5 *slots*, completo.

6.1.3. Conclusiones sobre el estudio del efecto de las retransmisiones realizado para el perfil SPP

El objetivo de este estudio ha sido evaluar, para el perfil SPP, el impacto de la ocurrencia de retransmisiones que se producen a nivel BB sobre el retardo de transmisión, independientemente de la naturaleza de la interferencia que provoca una tasa BER no nula.

En este caso, se ha comenzado proponiendo una cota superior para el retardo de transmisión, que se ha expresado en función de la tasa PER, calculada a partir de la tasa BER, suponiendo que las pérdidas son incorreladas. Es decir, se ha supuesto que la probabilidad de que haya i errores en la transmisión de un paquete se puede caracterizar mediante una distribución binomial.

Posteriormente, se ha validado el modelo, también mediante caracterización empírica. Para ello, durante el procedimiento de medida, al finalizar las

1000 repeticiones que se ejecutan para cada longitud de los datos de usuario, se ha obtenido un parámetro relativo a la calidad del enlace, proporcionado por el módulo Bluetooth, a partir del que se puede estimar la tasa BER, de acuerdo con una relación especificada por el fabricante.

Finalmente, el promedio de las tasas BER obtenidas para el rango completo de longitudes se ha incorporado en el modelo teórico, a fin de poder comparar el retardo teórico con el medido.

Como conclusión, cabe destacar que, pese a que el entorno de medida no es controlado y a que la precisión de la tasa BER estimada, como indica el fabricante no es buena, el retardo previsto teóricamente se ajusta, en gran medida, al obtenido experimentalmente.

6.2. Líneas futuras

Las líneas que se proponen para continuar investigando pueden clasificarse en tres grupos:

- Ampliación correspondiente a las nuevas características soportadas por Bluetooth 2.0 + EDR, que conlleva:
 - Modificar las ecuaciones que definen las componentes $t_{TX}(N)$ y $t_{ACK}(N)$ para que el cálculo del retardo de transmisión contemple los paquetes definidos en el modo EDR de Bluetooth 2.0: 2-DH1, 3-DH1, 2-DH3, 3-DH3, 2-DH5 y 3-DH5.

- Incluir en el modelo los modos de funcionamiento especificados en Bluetooth 2.0 + EDR para el nivel L2CAP: control de flujo y retransmisiones.
- Extensión para el funcionamiento en *scatternets* utilizando el perfil SPP: Se recomienda abordar esta línea en dos etapas, reformulando las ecuaciones del retardo extremo a extremo cuando las transmisiones se realizan:
 - Entre dos dispositivos de una *piconet*, cuando uno de ellos, maestro o esclavo, participa en otra *piconet*. El objetivo es comenzar estudiando cómo afecta al rendimiento el hecho de que la unidad que está conectada a las dos *piconets* tenga que recurrir a modos de bajo consumo para compartir el tiempo entre ambas subredes.
 - Entre dos dispositivos de una *scatternet*. En este caso, se sugiere empezar investigando el comportamiento de una *scatternet* básica, integrada por dos *piconets*, en la que participen únicamente tres dispositivos: un esclavo conectado a los maestros de cada una de las *piconets*. El objetivo, sería esbozar un modelo básico que iría completándose de forma incremental, conforme se vaya aumentando el número de unidades Bluetooth.
- Ampliaciones específicas del perfil PAN: 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.

Apéndice A

Medidas realizadas con Bluetooth 1.2 y Bluetooth 2.0

En primer lugar se han realizado medidas de retardo de transmisión en conexiones maestro-esclavo y esclavo-maestro utilizando dos dispositivos conformes a la especificación Bluetooth 2.0, de los fabricantes Ericsson [Ericsson] y Zaapa [Zaapa], los cuales incorporan circuitos integrados de CSR [CSR], con las siguientes versiones y revisiones del *firmware* HCI y del nivel LM, obtenidas, análogamente a como se indicó en el capítulo 5, es decir, con un comando facilitado por BlueZ:

- *Versión de HCI: 2.0; Revisión de HCI: 0x7a6*
- *Versión del nivel LM: 2.0; subversión del nivel LM: 0x706*

En este caso, es destacable que pese a que se configure explícitamente el uso de paquetes DH1, DH3 y DH5, sucede que, al interoperar dos dispositivos

Bluetooth 2.0, transmiten utilizando el modo EDR. Esta característica de funcionamiento provoca que los escalones equiespaciados que se producían en el retardo de transmisión cada vez que se llenaba un paquete de 5 *slots*, para Bluetooth 1.1, no se correspondan con los observados en las transmisiones Bluetooth 2.0 + EDR. Asimismo, EDR también causa que las pendientes de crecimiento en los retardos de transmisión sean distintas. Ambas diferencias de comportamiento pueden observarse en las gráficas representadas en la figura A.1, en la que se muestra el retardo de transmisión para intervalos de *poll* de 20 ms y 25 ms, en ambos sentidos de transmisión, obtenido en conexiones establecidas entre dispositivos Bluetooth 2.0 + EDR, en comparación con el medido para Bluetooth 1.1 y que se presentó en los apartados 5.3.1.1 y 5.3.1.2.

La justificación de este comportamiento requiere tener en cuenta, que para el modo EDR, se definen nuevos tipos de paquetes, concretamente, 2-DH1, 3-DH1, 2-DH3, 3-DH3, 2-DH5 y 3-DH5 que, aunque ocupan el mismo número de *slots* que los paquetes DH1, DH3 y DH5, se caracterizan porque se transmiten a mayor velocidad: 2 Mbps ó 3 Mbps, según el prefijo sea 2 ó 3, siendo capaces de transportar, por tanto, una mayor cantidad de información en 1, 3 ó 5 *slots*.

Así, teniendo como referencia la figura A.1 y la tabla A.1, en la que se indica la máxima capacidad de cada uno de los paquetes definidos para el modo EDR, puede deducirse que el funcionamiento es el siguiente: el nivel BB usará un paquete 2-DH1 mientras el número de octetos a transmitir no exceda su capacidad, es decir, los 54 octetos; en ese momento, se pasará a utilizar un paquete 3-DH1, hasta que se alcancen los 83 octetos, longitud a partir de la que se empleará el paquete 2-DH3, continuando, para longitudes superiores a los 367 octetos, con un paquete 3-DH3; cuando el tamaño supere los 552 octetos

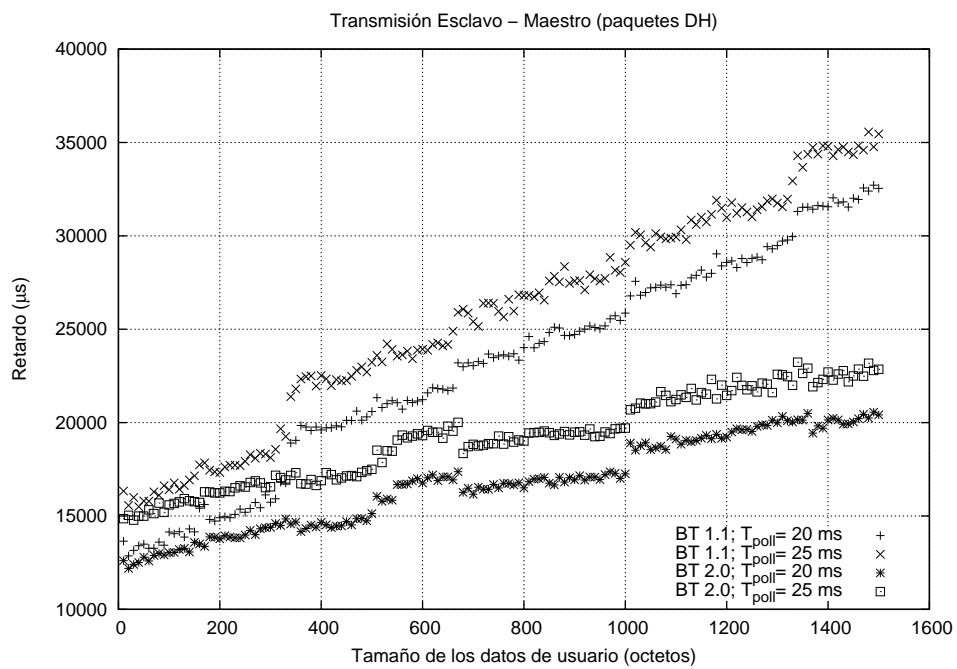
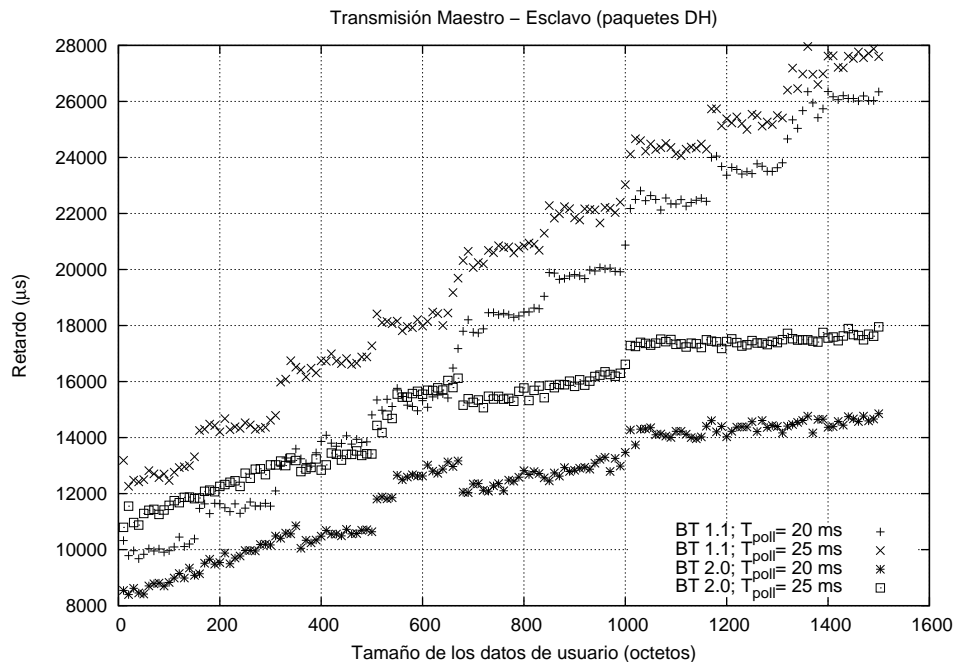


Figura A.1: Comparación de los retardos de transmisión obtenidos utilizando dispositivos Bluetooth 1.1 y Bluetooth 2.0 para diferentes valores del parámetro T_{poll}

Tipo de paquete ACL (EDR)	Tamaño (octetos)
2-DH1	0-54
2-DH3	0-367
2-DH5	0-679
3-DH1	0-83
3-DH3	0-552
3-DH5	0-1021

Tabla A.1: Tipos de paquetes ACL definidos en Bluetooth 2.0 + EDR

el nivel BB usará un paquete de tipo 2-DH5, aunque sólo hasta un tamaño de 679 octetos, longitud a partir de la que los datos se transmitirán en un paquete 3-DH5, que es capaz de transportar un total de 1017 octetos, correspondientes a 1008 octetos de datos de usuario más el *overhead* introducido por RFCOMM y L2CAP.

Con la finalidad de mostrar que el comportamiento real se ajusta a este esquema, en la figura A.2 se ha representado la curva del retardo medida para $T_{poll} = 25\text{ ms}$ y líneas verticales que marcan las longitudes 367 octetos, 552 octetos y 679 octetos, que representan los siguientes cambios de tipos de paquetes: 2-DH3 \rightarrow 3-DH3, 3-DH3 \rightarrow 2-DH5 y 2-DH5 \rightarrow 3-DH5. También se ha trazado otra línea a la longitud de 1021 octetos, capacidad máxima de un paquete 3-DH5, para resaltar que el escalón que se produce a este tamaño, se debe a que, para longitudes superiores, habrá que esperar hasta 6 *slots* antes de proseguir enviando fragmentos de continuación. Nótese que hasta este punto, ha sido suficiente la utilización de un único paquete.

En definitiva, la selección del tipo de paquete que lleva a cabo el nivel BB de un módulo Bluetooth 2.0, al comunicarse con otra unidad Bluetooth 2.0, se rige

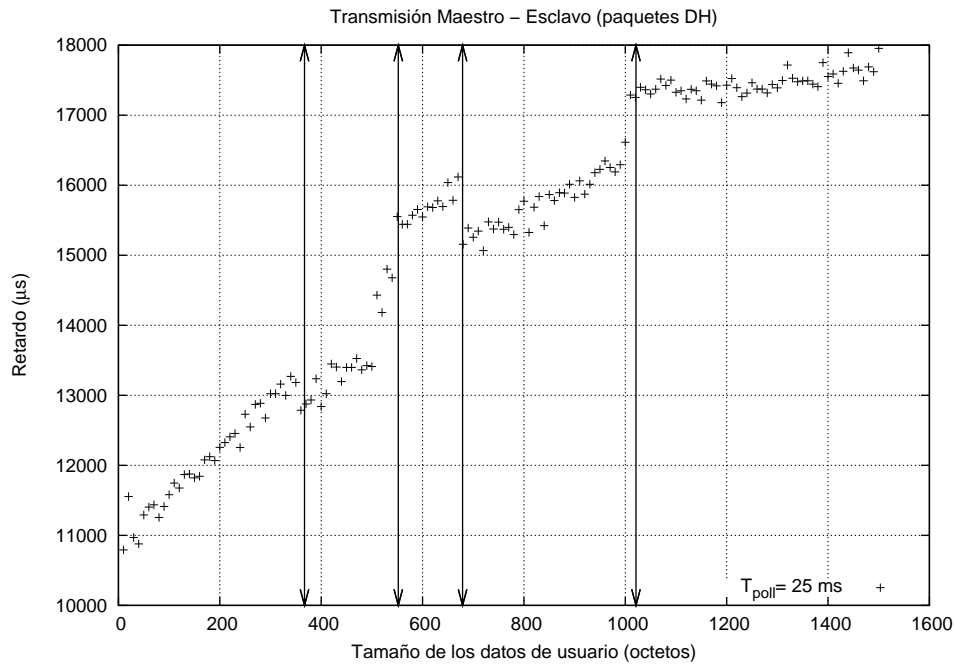


Figura A.2: Curva del retardo de transmisión obtenida en el modo EDR de Bluetooth 2.0 para un valor del parámetro T_{poll} de 25 ms, en la que se han marcado, con líneas verticales, las longitudes 367 octetos, 552 octetos, 679 octetos y 1021 octetos.

por los siguientes criterios:

- Antes de recurrir a un paquete que ocupe un mayor número de *slots*, el nivel BB opta por aumentar la velocidad de transferencia, hecho que da lugar a un escalón descendente. En este caso, longitudes de datos de usuario mayores experimentan un menor retardo de transmisión.
- Cuando se excede la capacidad de un paquete 3-DH, el nivel BB sí utiliza un paquete que se extienda a lo largo de más *slots*, pero reduce la velocidad de transferencia. Esto es, el paquete seleccionado pasa a ser de tipo 2-DH.

Adicionalmente, dado que el modo EDR no permite constatar si el modelo definido en la presente tesis para Bluetooth 1.1 es igualmente válido para Bluetooth 2.0, se ha optado por repetir las medidas para una *piconet* de dos

dispositivos, reemplazando un módulo Bluetooth 2.0 por uno conforme a Bluetooth 1.2 del fabricante Tvisto [Tvisto], que incorpora un circuito integrado del fabricante *Integrated System Solution Corp* [ISSC], con las versiones y revisiones del *firmware* HCI y del nivel LM, que a continuación se indican:

- *Versión de HCI: 1.2; Revisión de HCI: 0x1fe*
- *Versión del nivel LM: 1.2; subversión del nivel LM: 0x1fe*

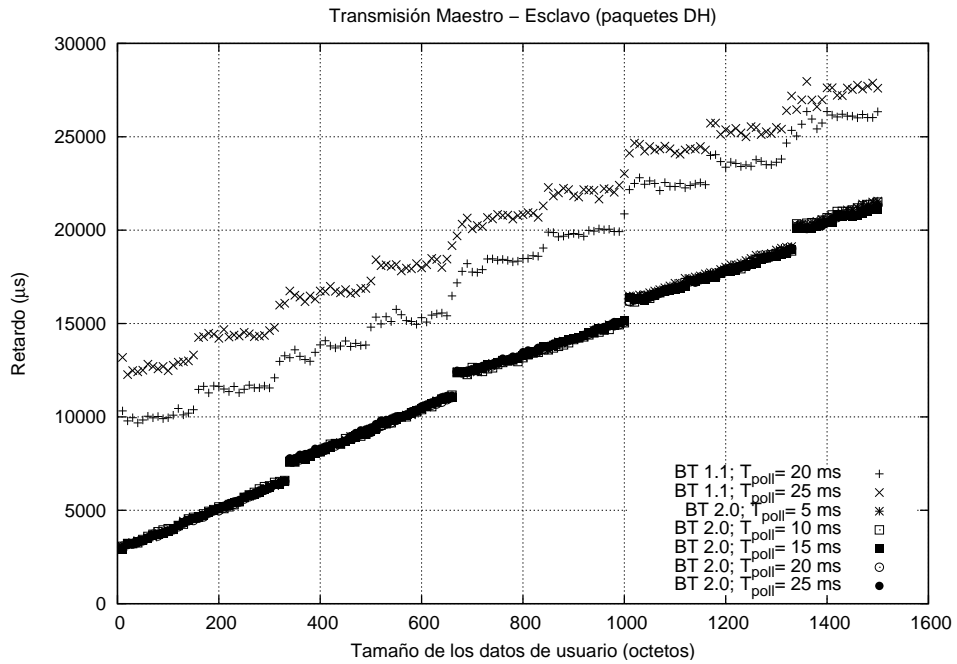
De este modo, se garantiza que el dispositivo Bluetooth 2.0 opere en el modo básico.

El resultado de las pruebas realizadas permite concluir no sólo que el modelo es válido sino que, además, las correcciones introducidas para Bluetooth 1.1 son innecesarias. Concretamente, en el sentido esclavo-maestro desaparece el *offset* de 2 ms que estaba presente independientemente del valor del intervalo de *poll* configurado (véase el apartado 5.3.1.2), de forma que todos los retardos medidos, según se ilustra en la figura A.3 (a), están desplazados $T_{poll}/2$. El valor $T_{poll}/2$, como ya se explicó, es la media de una variable aleatoria que sigue una distribución uniforme entre 0 y T_{poll} , y se corresponde con el tiempo medio transcurrido desde que el *Host* solicita el envío de datos, hasta que el nivel BB transmita esos datos.

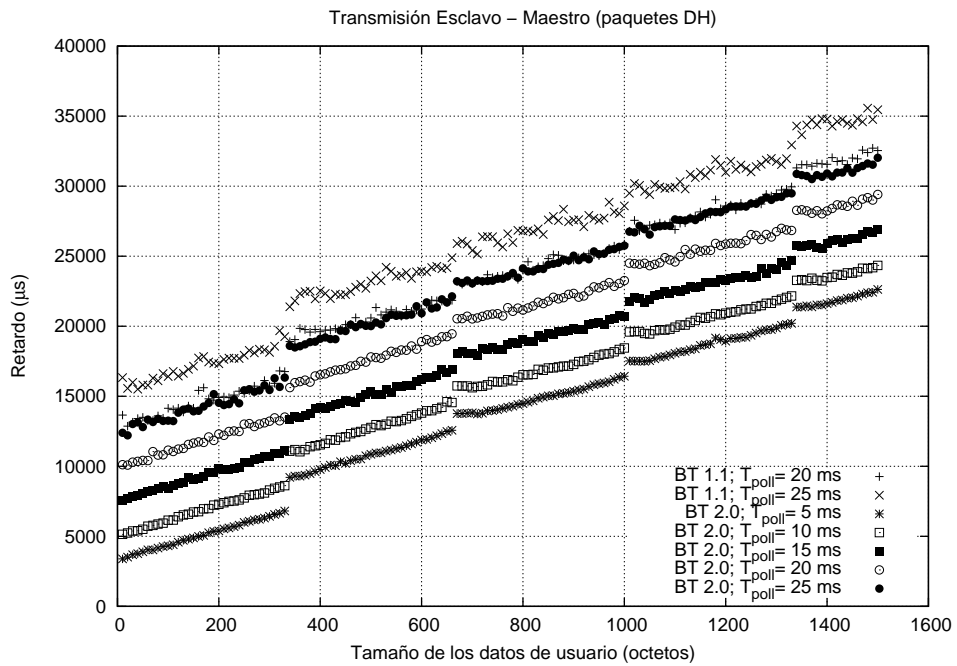
En el sentido de transmisión maestro-esclavo, la gráfica representada en la figura A.3 (b) pone de manifiesto que el retardo medido es independiente del valor de T_{poll} configurado. También, los resultados obtenidos en este caso, revelan que puede prescindirse de la corrección que se introdujo en el apartado 5.3.1.4 para modelar los escalones debidos a que, en Bluetooth 1.1, el

esclavo no entregaba los datos al *Host* (PC) hasta que no transcurriera, al menos, el tiempo correspondiente a un paquete de 1, 3 ó 5 *slots*, completo.

Además, con la intención de facilitar la comparación de las curvas de retardo obtenidas con las que se presentaron en la sección 5.3, en la figura A.3 se han incluido los retardos medidos en Bluetooth 1.1, para $T_{poll} = 20\text{ ms}$ y $T_{poll} = 25\text{ ms}$. Finalmente, en la figura A.4 se ha representado el retardo analítico medio calculado para $T_{poll} = 10\text{ ms}$, y el medido para todos los valores de T_{poll} en el sentido maestro-esclavo. En lo que respecta a las transmisiones del esclavo, por motivos de legibilidad, los retardos medidos y analíticos se han representado, individualmente, en las figuras A.5, A.6 y A.7. Puede observarse que el ajuste es prácticamente perfecto. No obstante, nótese que la corrección que se aplicó en el apartado 5.3.1.5 para contemplar el hecho que la confirmación del primer paquete de cinco *slots* no es efectuada por el maestro en el *slot* inmediatamente posterior a la recepción parece resultar excesiva en algunos casos, como en los representados para $T_{poll} = 10\text{ ms}$, $T_{poll} = 15\text{ ms}$ y $T_{poll} = 20\text{ ms}$, aunque sólo para las longitudes comprendidas entre los 330 octetos y los 660 octetos.



(a) Sentido de transmisión maestro-esclavo



(b) Sentido de transmisión esclavo-maestro

Figura A.3: Comparación de los retardos de transmisión obtenidos utilizando dispositivos Bluetooth 1.1 y Bluetooth 2.0 para diferentes valores del parámetro T_{poll}

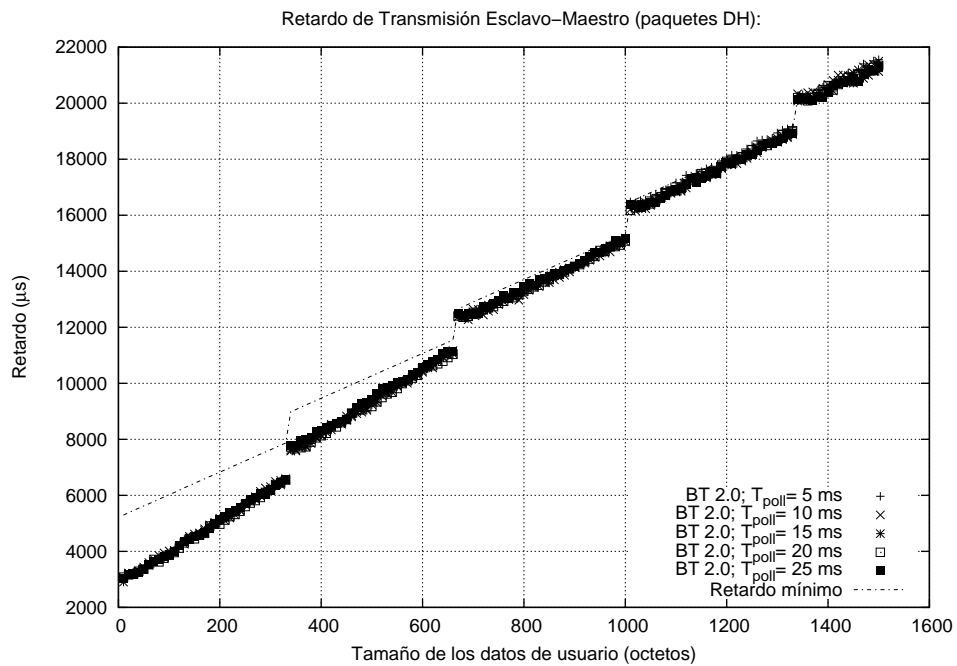
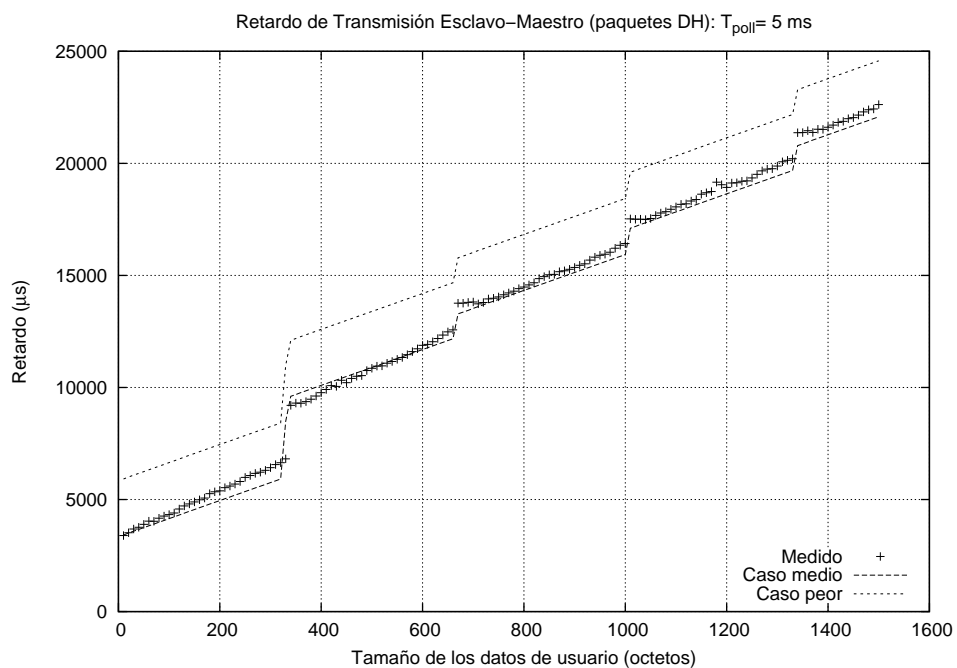


Figura A.4: Comparación de los retardos de transmisión maestro-esclavo medido y teórico para valores del parámetro T_{poll} de 5 ms, 10 ms, 15 ms, 20 ms y 25 ms



(a)

Figura A.5: Comparación de los retardos de transmisión esclavo-maestro medido y teórico para valores del parámetro T_{poll} de 5 ms

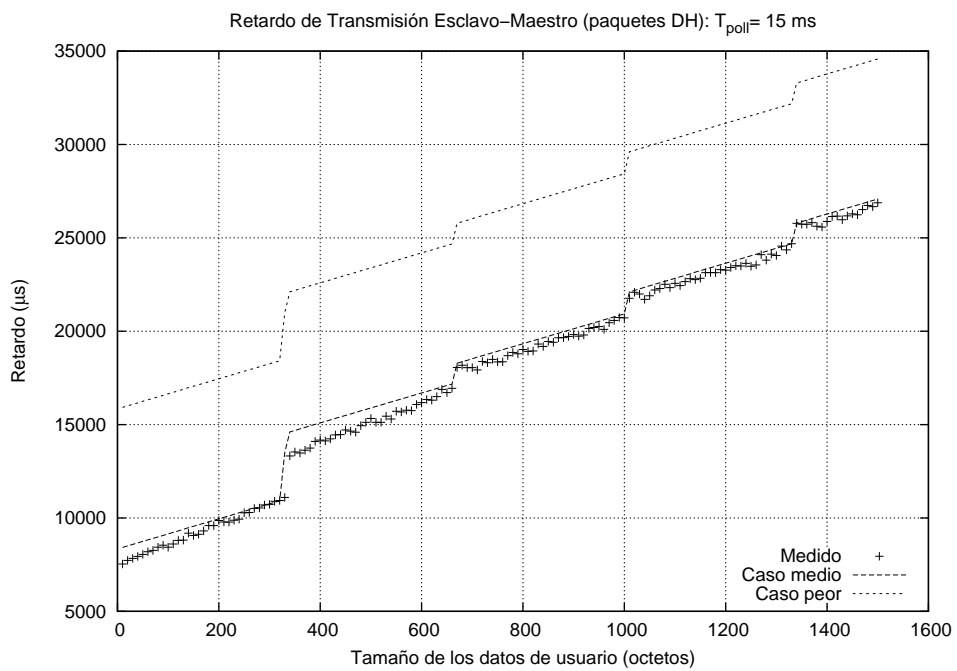
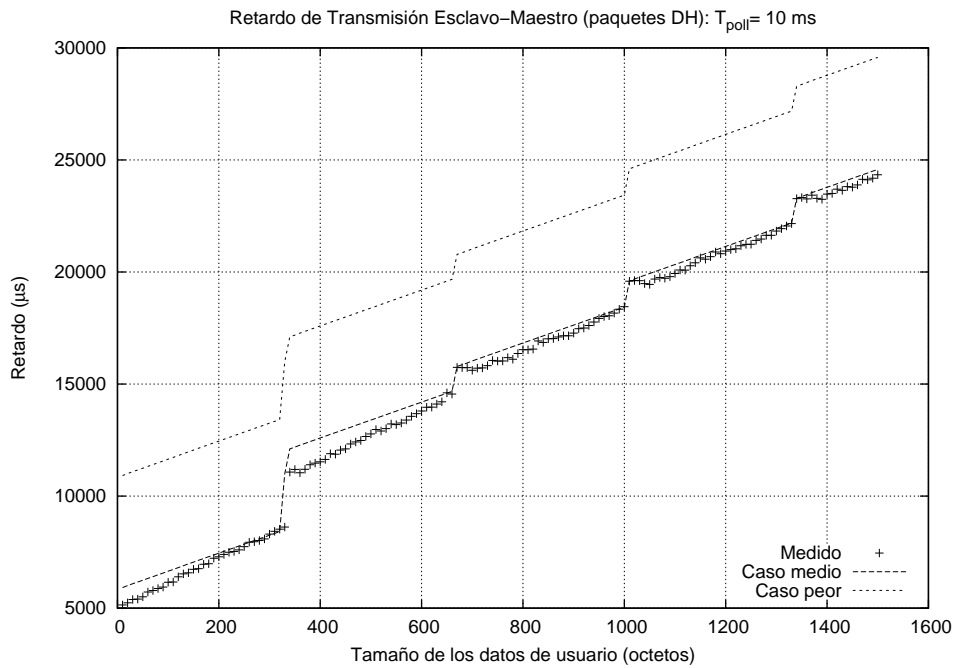


Figura A.6: Comparación de los retardos de transmisión esclavo-maestro medido y teórico para valores del parámetro T_{poll} de 10 ms y 15 ms

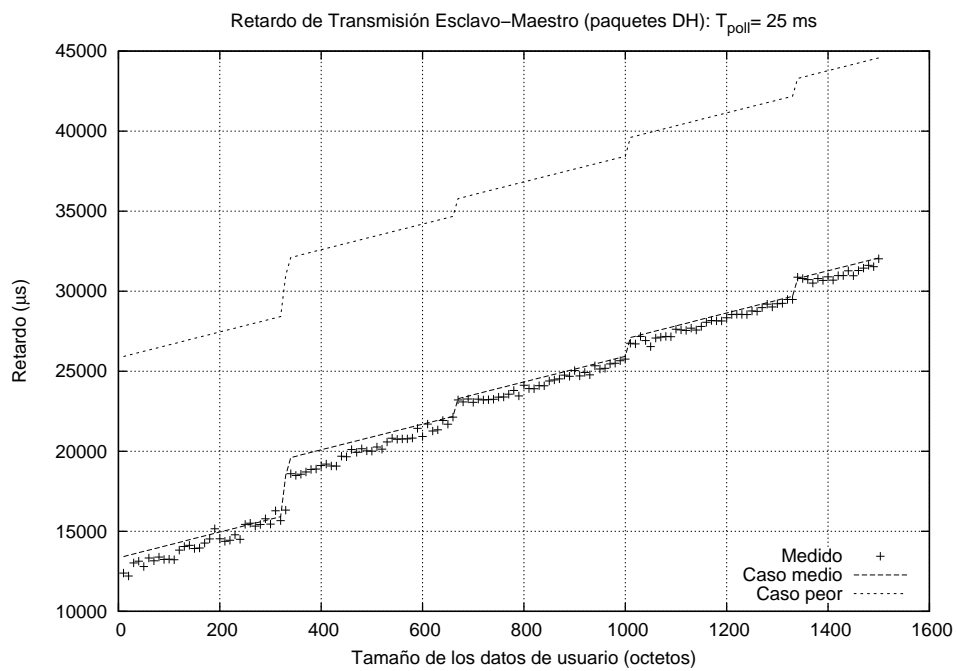
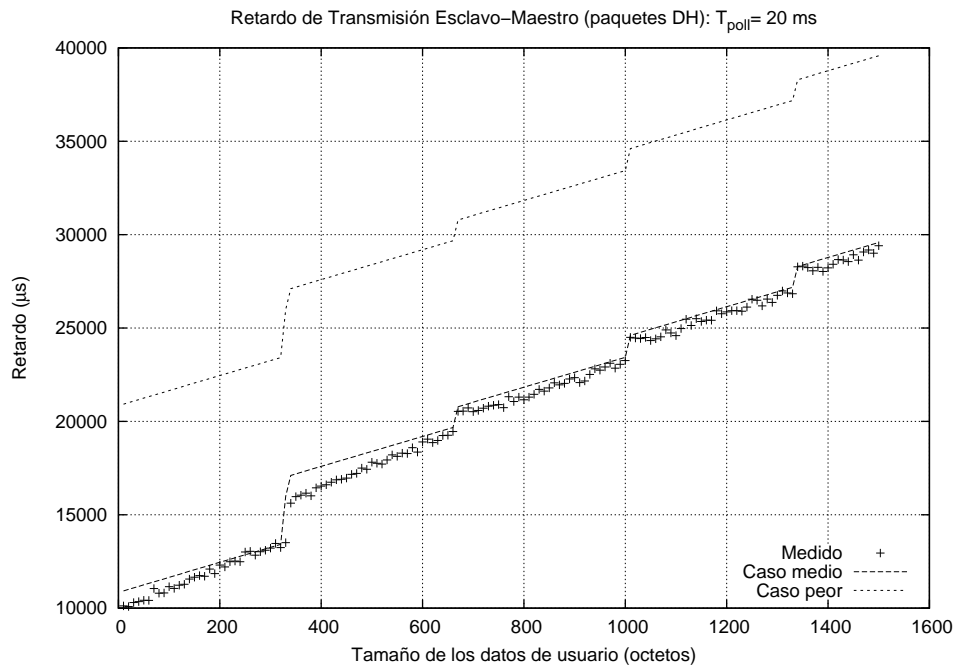


Figura A.7: Comparación de los retardos de transmisión esclavo-maestro medido y teórico para valores del parámetro T_{poll} de 20 ms y 25 ms

Bibliografía

- [A&D Medical] A&D MEDICAL: Internet.
URL <http://www.andmedical.com.au/>
- [Ali05] ALI, R., y J. POLLARD: «Real-time Voice over IP over Bluetooth», en *Proceedings of the Third Workshop IEEE Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Sept. 2005, págs. 580–583.
- [Ashraf06] ASHRAF, I., A. GKELIAS, M. DOHLER, y A. AGHVAMI: «Time-synchronised multi-piconet Bluetooth environments», *IEE Proceedings-Communications*, tomo 153, nº 3, 2006, págs. 445–452.
- [Bell03] BELL, D., J.-C. CANO, y P. MANZONI: «Evaluating Bluetooth performance as the support for context-aware applications», en *Proceedings of the 12th International Conference on Computer Communications and Networks*, 20-22 Oct. 2003, págs. 345–350.
- [Bluetooth SIG01a] BLUETOOTH SIG: «Bluetooth Assigned Numbers Version 1.1», en *Specification of the Bluetooth System*, Bluetooth Special

Interest Group, 22 February 2001.

URL <http://www.bluetooth.com/>

[Bluetooth SIG01b] BLUETOOTH SIG: «Bluetooth continuous variable slope delta (CVSD) encoded test signal Version 2.1», en *Specification of the Bluetooth System*, Bluetooth Special Interest Group, 22 February 2001.

URL <http://www.bluetooth.com/>

[Bluetooth SIG01c] BLUETOOTH SIG: «Bluetooth Core Specification Version 1.1», en *Specification of the Bluetooth System*, tomo 1, Bluetooth Special Interest Group, 22 February 2001.

URL <http://www.bluetooth.com/>

[Bluetooth SIG01d] BLUETOOTH SIG: «Bluetooth Network Encapsulation Protocol (BNEP) Specification Revision 0.95a», Bluetooth Special Interest Group, 22 June 2001.

URL <http://www.bluetooth.com/>

[Bluetooth SIG01e] BLUETOOTH SIG: «Bluetooth Personal Area Networking Profile Revision 0.95a», Bluetooth Special Interest Group, 26 June 2001.

URL <http://www.bluetooth.com/>

[Bluetooth SIG01f] BLUETOOTH SIG: «Bluetooth Profiles Specification Version 1.1», en *Specification of the Bluetooth System*, tomo 2, Bluetooth Special Interest Group, 22 February 2001.

URL <http://www.bluetooth.com/>

[Bluetooth SIG03a] BLUETOOTH SIG: «Bluetooth Core Specification Version

- 1.2», en *Specification of the Bluetooth System*, Bluetooth Special Interest Group, 2003.
URL <http://www.bluetooth.com/>
- [Bluetooth SIG03b] BLUETOOTH SIG: «Bluetooth Network Encapsulation Protocol (BNEP) Specification Revision 1.0», Bluetooth Special Interest Group, 14 February 2003.
URL <http://www.bluetooth.com/>
- [Bluetooth SIG04] BLUETOOTH SIG: «Bluetooth Core Specification Version 2.0+EDR», en *Specification of the Bluetooth System*, Bluetooth Special Interest Group, 2004.
URL <http://www.bluetooth.com/>
- [Bluetooth SIG07] BLUETOOTH SIG: «Bluetooth Core Specification Version 2.1+EDR», en *Specification of the Bluetooth System*, Bluetooth Special Interest Group, 2007.
URL <http://www.bluetooth.com/>
- [BlueZ06] BLUEZ: «BlueZ Project: BlueZ, Bluetooth Protocol Stack for Linux», 2006.
URL <http://www.bluez.org/>
- [Chan04] CHAN, K. L., V. MISIC, y J. MISIC: «Efficient polling schemes for Bluetooth picocells revisited», en *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, 5-8 Jan. 2004, págs. 307–314.
- [Chen04] CHEN, L.-J., R. KAPOOR, M. SANADIDI, y M. GERLA: «Enhancing Bluetooth TCP throughput via link layer packet

- adaptation», en *Proceedings of IEEE International Conference on Communications*, tomo 7, 20-24 June 2004, págs. 4012–4016.
- [Chen05] CHEN, Y.-S., K.-S. LIU, y Y.-W. LIN: «A credit-based on-demand QoS routing protocol over Bluetooth WPANs», en *Proceedings of 10th IEEE Symposium on Computers and Communications*, 27-30 June 2005, págs. 807–812.
- [Conceptronic] CONCEPTRONIC: Intenet.
URL <http://www.conceptronic.net/>
- [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, 2003, págs. 259–269.
- [CorScience] CORSCIENCE: Internet.
URL <http://www.corscience.de/>
- [Corvaja06] CORVAJA, R.: «QoS Analysis in Overlay Bluetooth-WiFi Networks with Profile-Based Vertical Handover», *IEEE Transactions on Mobile Computing*, tomo 5, nº 12, December 2006, págs. 1679–1690.
- [CSR] CSR: «Cambridge Silicon Radio Plc. BlueCore Bluetooth chipset.», Intenet.
URL <http://www.csr.com/products/bcrange.htm>

-
- [D-Link] D-LINK: Intenet.
URL <http://www.d-link.es/>
- [Das01] DAS, A., A. GHOSE, A. RAZDAN, H. SARAN, y R. SHOREY: «Enhancing performance of asynchronous data traffic over the Bluetooth wireless ad-hoc network», en *Proceedings of IEEE 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, tomo 1, Anchorage, AK, 2001, págs. 591–600.
- [Ericsson] ERICSSON: Intenet.
URL <http://www.ericsson.com/>
- [ETSI97] ETSI: *ETSI TS 101 369:Digital cellular telecommunications system (Phase 2+); Terminal Equipment to Mobile Station (TEMS) multiplexer protocol (3PP TS 07.10 version 6.5.0 Release 1997)*, 1997.
URL <http://www.etsi.org/>
- [Feng04] FENG, W., A. NALLANATHAN, y H. GARG: «Performance of PHY and MAC layers of a Bluetooth piconet in multi-Bluetooth interference environment», en *Proceedings of IEEE Global Telecommunications Conference*, tomo 6, 29 Nov.-3 Dec. 2004, págs. 3614–3618.
- [Francia04] FRANCIA, G. A., A. KILARU, L. PHUONG, y M. VASHI: «An empirical study of Bluetooth performance», en *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.
- [Golmie01a] GOLMIE, N., N. CHEVROLIER, y I. ELBAKKOURI: «Interference aware Bluetooth packet scheduling», en *Proceedings of IEEE Global Telecommunications Conference*, tomo 5, 25-29 Nov. 2001, págs. 2857–2863.
- [Golmie01b] GOLMIE, N., R. E. V. DYCK, y A. SOLTANIAN: «Interference of bluetooth and IEEE 802.11: simulation modeling and performance evaluation», en *Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, ACM, New York, NY, USA, 2001, págs. 11–18.
- [Golmie03a] GOLMIE, N., N. CHEVROLIER, y O. REBALA: «Bluetooth and WLAN coexistence: challenges and solutions», *IEEE Wireless Communications*, tomo 10, nº 6, 2003, págs. 22–29.
- [Golmie03b] GOLMIE, N., R. E. V. DYCK, A. SOLTANIAN, A. TONNERRE, y O. RÉBALA: «Interference evaluation of Bluetooth and IEEE 802.11b systems», *Wireless Networks*, tomo 9, nº 3, 2003, págs. 201–211.
- [Golmie03c] GOLMIE, N., O. REBALA, y N. CHEVROLIER: «Bluetooth adaptive frequency hopping and scheduling», en *Proceedings of IEEE Military Communications Conference*, tomo 2, 13-16 Oct. 2003, págs. 1138–1142.
- [Golmie04] GOLMIE, N.: «Bluetooth dynamic scheduling and interferen-

-
- ce mitigation», *Mobile Networks and Applications*, tomo 9, nº 1, 2004, págs. 21–31.
- [Howitt03] HOWITT, I., y S. Y. HAM: «Site specific WLAN and WPAN coexistence evaluation», en *Proceedings of IEEE Wireless Communications and Networking*, tomo 3, 16-20 March 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 *Proceedings of Military Communications Conference*, 23-25 Oct. 2006, págs. 1–7.
- [Huang05] HUANG, P., y A. C. BOUCOUVALAS: «Delay Analysis for Bluetooth Baseband ACL Packets», en *Proceedings of Convergence of Telecommunications, Networking & Broadcasting Symposium*, Liverpool, 2005, págs. 396–401.
- [IEEE 1073 96] IEEE 1073 : *IEEE Std 1073-1996. IEEE Standard for Medical Device Communications—Overview and Framework*, 1996.
URL <http://ieeexplore.ieee.org/servlet/opac?punumber=4130>
- [IEEE 1073 07] IEEE 1073 : *IEEE Unapproved Std P11073-00101/D02J, Feb 2007. Unapproved Draft Guide for Health informatics – Point-of-care medical device communication – Technical report – Guidelines for the use of RF wireless technology*, 2007.
URL <http://ieeexplore.ieee.org/servlet/opac?punumber=4152658>

- [IEEE 802] IEEE 802: «IEEE 802 LAN/MAN Standards Committee», Internet.
URL <http://www.ieee802.org/>
- [IEEE 802 02] IEEE 802 : *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)*, 2002.
URL <http://ieeexplore.ieee.org/servlet/opac?punumber=7932>
- [IEEE 802 03a] IEEE 802 : *IEEE Std 802.15.3 - 2003 IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs)*, 2003.
URL <http://ieeexplore.ieee.org/servlet/opac?punumber=8821>
- [IEEE 802 03b] IEEE 802 : *IEEE Std 802.15.4 - 2003 IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*,

2003.
URL <http://ieeexplore.ieee.org/servlet/opac?punumber=8762>
- [IEEE 802 05] IEEE 802 : *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)*, 2005.
URL <http://ieeexplore.ieee.org/servlet/opac?punumber=9980>
- [IEEE 802 06] IEEE 802 : *IEEE Std 802.15.4 - 2006 IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, 2006.
URL <http://ieeexplore.ieee.org/servlet/opac?punumber=11161>
- [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>

- [ISSC] ISSC: «Integrated System Solution Corp.», Intenet.
URL <http://www.issc.com.tw/>
- [Jiang05] JIANG, Y., Z. JIN, S. MENG, Y. SHU, y L. ZHANG: «Enhanced polling mechanism of Bluetooth for realtime traffic», en *Proceedings of Canadian Conference on Electrical and Computer Engineering*, 1-4 May 2005, págs. 1180–1183.
- [Khoutaif06] KHOUTAIF, T., y G. JUANOLE: «Formal modelling and evaluation of the data transfer phase of the ACL links on the WPAN Bluetooth», en *Proceedings of IEEE Conference on Emerging Technologies and Factory Automation*, 20-22 Sept. 2006, págs. 30–37.
- [Latkoski07] LATKOSKI, P., y L. GAVRILOVSKA: «Analysis of WWW performance in IEEE 802.15 based systems», en *Proceedings of the 8th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services*, 26-28 Sept. 2007, págs. 95–98.
- [Li07a] LI, J., y X. LIU: «Evaluation of co-channel and adjacent channel interference for multiple Bluetooth piconets with dual channel transmission», en *Proceedings of IEEE Wireless Communications and Networking Conference*, 11-15 March 2007, págs. 2355–2360.
- [Li07b] LI, J., X. LIU, y X. MA: «Dual channel transmission for coexistence of Bluetooth piconets with multi-slot packets»,

-
- en *Proceedings of 2nd International Symposium on Wireless Pervasive Computing*, 2007.
- [Linksys] LINKSYS: Intenet.
URL <http://www.es-linksys.com/>
- [M. Cordeiro04] DE M. CORDEIRO, C., 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, Aug 2004, págs. 409–422.
- [Ma05] MA, M., y S. Y. LOW: «Supporting real-time service in Bluetooth networks», en *Proceedings of Workshop on High Performance Switching and Routing*, 12-14 May 2005, págs. 167–171.
- [Madhavapeddy05] MADHAVAPEDDY, A., y A. TSE: «A study of Bluetooth propagation using accurate indoor location mapping», en *Proceedings of UbiComp 2005: Ubiquitous Computing*, tomo 3660/2005 de *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2005, págs. 105–122.
- [Mistic04] MISIC, V., E. KO, y J. MISIC: «Load and QoS-adaptive scheduling in Bluetooth piconets», en *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, 2004.
- [Moon05] MOON, T. K.: *Error Correction Coding: Mathematical Methods and Algorithms*, Wiley-Interscience, 2005.

- [Moron05a] MORON, M., E. CASILARI, y J. GAZQUEZ: «Sistema de Monitorización Inalámbrica de sensores de SPO2 (pulsioxímetros)», en *Libro de Ponencias Mundo Internet*, Madrid, 2005, págs. 96 – 102.
- [Moron05b] MORON, M., E. CASILARI, R. LUQUE, y J. GAZQUEZ: «A wireless monitoring system for pulse-oximetry sensors», en *Proceedings Of Systems Communications, Simposium on Advanced Industrial Conference on Wireless Technologies ICW'2005*, Montreal, Canada, 14-17 Aug. 2005, págs. 79–84.
- [Moron05c] MORON, M., J. LUQUE, E. CASILARI, J. GAZQUEZ, y A. DIAZ-ESTRELLA: «Monitorización inteligente de pacientes mediante tecnologías inalámbricas», en *Actas del I Simposio sobre Computación Ubicua e Inteligencia Ambiental (UCAMI 2005)*, Granada, Septiembre 2005, págs. 313 – 320.
- [Moron07a] MORON, M., J. LUQUE, A. BOTELLA, E. CUBEROS, E. CASILARI, y A. DIAZ-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.
- [Moron07b] MORON, M., R. LUQUE, E. CASILARI, y A. DIAZ-ESTRELLA: «An Analytical Study of the Delay in Bluetooth Networks Using the Personal Area Network Profile», *IEEE Communi-*

-
- cations Letters*, tomo 11, nº 11, November 2007, págs. 845–847.
- [Moron07c] MORON, M. J., J. R. LUQUE, A. A. BOTELLA, E. J. CUBEROS, E. CASILARI, y A. DIAZ-ESTRELLA: «J2ME and smart phones as platform for a Bluetooth Body Area Network for Patient-telemonitoring», en *Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 22-26 Aug. 2007, págs. 2791–2794.
- [Moron08] MORON, M. J., R. LUQUE, E. CASILARI, y A. DIAZ-ESTRELLA: «Analysis of Bluetooth Transmission Delay in Personal Area Networks», en *Proceedings of the IEEE International Symposium on Wireless Pervasive Computing (ISWPC 2008)*, Santorini (Greece), 7–9 May 2008, págs. 620–622.
- [Nonin Medical] NONIN MEDICAL: Internet.
URL <http://www.nonin.com/>
- [NS-2] NS-2: «Network Simulator (NS-2)», Internet.
URL <http://www.isi.edu/nsnam/ns/>
- [Okura02] OKURA, H., M. KATO, y S. TASAKA: «The effect of segmentation mismatch on quality of continuous media transmission by Bluetooth», en *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, tomo 2, 2002, págs. 702–709.
- [Opnet] OPNET: «Opnet Modeler», Internet.

- URL http://www.opnet.com/solutions/network_rd/modeler.html
- [Pasolini03] PASOLINI, G.: «Bluetooth piconets coexistence: analytical investigation on the optimal operating conditions», en *Proceedings of IEEE International Conference on Communications*, tomo 1, 11-15 May 2003, págs. 198–202.
- [Pasolini04] PASOLINI, G. .: «Analytical investigation on the coexistence of Bluetooth piconets», *IEEE Communications Letters*, tomo 8, nº 3, 2004, págs. 144–146.
- [Peterson03] PETERSON, B., R. BALDWIN, J. KHAROUFEH, y R. RAINES: «Refinements to the packet error rate upper bound for Bluetooth networks», *IEEE Communications Letters*, tomo 7, nº 8, 2003, págs. 382–384.
- [Shinde05] SHINDE, H., y M. BORSE: «High-rate wireless personal area networks [multimedia capable]», en *Proceedings of IEEE International Conference on Personal Wireless Communications*, 23-25 Jan. 2005, págs. 19–23.
- [Sitecom] SITECOM: Intenet.
URL <http://www.sitecom.com>
- [Tvisto] TVISTO: Intenet.
URL <http://www.t-visto.com>
- [Ubuntu Linux] UBUNTU LINUX: Intenet.
URL <http://www.ubuntu.com/>

-
- [Wang07] WANG, D., Y. LU, H. ZHANG, J. ZHANG, y Y. HAN: «Body Piconet for the Telemedicine System Based on Home Networks», en *Proceedings of IEEE/ICME International Conference on Complex Medical Engineering*, 23-27 May 2007, págs. 1202–1205.
- [Warren03] WARREN, S., J. YAO, R. SCHMITZ, y L. NAGL: «Wearable telemonitoring systems designed with interoperability in mind», en *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, tomo 4, 17-21 Sept. 2003, págs. 3736–3739.
- [Warren04] WARREN, S., J. YAO, R. SCHMITZ, y J. LEBAK: «Reconfigurable Point-of-Care Systems Designed with Interoperability Standards», en *Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, tomo 2, 2004, págs. 3270–3273.
- [Warren05] WARREN, S., J. LEBAK, J. CREEKMORE, A. MILENKOVIC, y E. JOVANOVIĆ: «Interoperability and Security in Wireless Body Area Network Infrastructures», en *Proceedings of the IEEE 27th Annual International Conference of the Engineering in Medicine and Biology Society*, 2005, págs. 3837–3840.
- [Wu04] WU, T.-H., C. KE, C. SHIEH, y W. HWANG: «A Practical Approach for Providing QoS in Bluetooth Piconet», en *Proceedings of the 3rd Annual Mediterranean Ad Hoc Networking Workshop*, Bodrum, Turkey, 2004.

- [Yao05] YAO, J., R. SCHMITZ, y S. WARREN: «A wearable point-of-care system for home use that incorporates plug-and-play and wireless standards», *IEEE Transactions on Information Technology in Biomedicine*, tomo 9, n^o 3, 2005, págs. 363–371.
- [Yen05] YEN, L.-H., y C.-H. LIAO: «Round-robin with FCFS preemption: a simple MAC scheduling scheme for Bluetooth piconet», en *Proceedings of 19th International Conference on Advanced Information Networking and Applications*, tomo 1, 28-30 March 2005, págs. 761–766.
- [Yong05] YONG, S. S., y W. H. KWON: «Performance analysis of Bluetooth piconets under mutual interference using bit error rate», en *Proceedings of International Network Conference*, Samos Island, Greece, jul 2005.
- [Zaapa] ZAAPA: Intenet.
URL <http://www.zaapa.co.uk>
- [Záruba04] ZÁRUBA, G. V., y P. JOHANSSON: «Guest Editorial: Advances in Research of Wireless Personal Area Networking and Bluetooth Enabled Networks», *Mobile Networks and Applications*, tomo 9, n^o 1, feb 2004, págs. 7–8.