Overhead and Segmentation Mismatch Effect on Bluetooth WPAN Performance

M. J. Moron · R. Luque · E. Casilari · A. Diaz-Estrella

Abstract Currently, Bluetooth is the most widely used technology for Wireless Personal Area Networks (WPAN). Quality-of-Service (QoS) support is critical to ensure bandwidth maximization for mobile applications based on this WPAN technology. The overhead introduced by the different layers of Bluetooth protocol may have a serious impact on WPAN performance. However, most studies of Bluetooth performance neglect this overhead and assume that data are directly transmitted over L2CAP (Logical Link Control and Adaptation Protocol) or even HCI (Host Controller Interface) layers. In fact, this option is not feasible in most Bluetooth applications, as they integrate actual devices that implement a particular Bluetooth profile, usually SPP (Serial Port Profile). The use of profiles cannot be disregarded as they guarantee the interoperability between devices from different vendors. The aim of this paper is to characterise the performance of a Bluetooth WPAN (specifically the end-to-end delay and the throughput) when profiles are utilised. This study takes into account the overhead added by the protocols taking part in the transmission of user data. This paper also explores the effect of segmentation mismatch that may appear when the maximum size for data in each layer of the architecture is different. The analysis has been focused on SPP and PAN (Personal Area Networks) profiles. In the case of the PAN profile, the study concludes that the network performance decreases for user data sizes greater than 1,472 bytes, since the excessive overhead added by the network layer is increased by the IP (Internet Proto-col) fragmentation. In the case of SPP, an inappropriate choice of the maximum data unit at RFCOMM (Radio Frequency Communication for Serial Cable Emulation Protocol based on ETSI TS 07.10) and L2CAP layers can also heavily affect the transmission delay.

M. J. Moron (✉) · R. Luque · E. Casilari · A. Diaz-Estrella
Departamento de Tecnologia Electronica, Universidad de Malaga, Malaga, Spain
e-mail: mjmoron@uma.es

R. Luque
e-mail: rluque@uma.es

E. Casilari
e-mail: ecasilari@uma.es

A. Diaz-Estrella
e-mail: adiaz@uma.es

Keywords WPAN · Bluetooth · PAN · SPP · Protocol · Overhead · Segmentation mismatch · End-to-end delay · Throughput

1 Introduction

WPANs (Wireless Personal Area Networks) are short range communication systems (from a few centimetres to about 10 m) designed to exchange information between devices within a small transmission area around a person. WPANs can be used to replace cables between PCs and peripherals with temporary or permanent connections, as well as to facilitate pervasive computing or the deployment of ad-hoc networks [10,16,20].

There are three categories of WPANs, as summarised in [10]: high-rate WPANs (802.15.3), medium-rate WPANs (802.15.1/Bluetooth) and low-rate WPANs (802.15.4). Nowadays Bluetooth is the most widely utilised technology for WPANs [16,20]. In fact, the IEEE 802 committee created the 802.15 workgroup (intended to standardize WPAN interfaces and protocols) upon realizing the importance of short-range wireless networks. This committee decided to adopt the lower layers of Bluetooth as part of the IEEE 802.15.1 standard [20].

Bluetooth [3] operates on the 2.4 GHz band, offering a bit rate of up to 1 Mbps. Bluetooth modules are characterised by low power consumption and small size. However, as noted in [20], the key feature that has enabled the progress of Bluetooth WPANs is the low cost of Bluetooth implementation, already integrated in a wide range of commercial devices, ranging from laptops, mobile phones and PDAs (Personal Digital Assistant) to medical sensors.

The interoperability is another critical factor to favour the expansion of any technology. As a means of guaranteeing the interoperation of devices from different providers, the Bluetooth specification defines the so-called profiles [4]. These profiles describe the requirements that a device must cover, as well as all the protocols and procedures to be implemented in different application scenarios, in order to make use of a particular service. It is worth mentioning the following profiles:

- GAP (*Generic Access Profile*): defines the generic procedures for the discovery of other devices and the corresponding connection setup.
- SDAP (*Service Discovery Application Profile*): focuses on the mechanisms that allow the discovery of services existing in other Bluetooth devices.
- PAN (*Personal Area Network*): describes how two or more Bluetooth devices can create an ad-hoc network, and the procedures to access remote networks through access points. The roles defined for this profile are NAP (*Network Access Point Profile*), GN (*Group Ad-hoc Network*), and PAN (*Personal Area Network User*).
- SPP (*Serial Port Profile*): defines the requirements for the communication between devices through serial port emulation using the RFCOMM protocol.
- DUN (*Dial-Up Networking Profile*): indicates how to provide access to remote network services by means of a modem connection emulation.
- LAN *Access Profile*: specifies the interconnection mechanisms between Bluetooth devices and Local Area Networks (LAN).

For a Bluetooth WPAN, the PAN profile [2] is the most appropriate option, especially if the objectives of the application include that:

- Devices belonging to the WPAN can be addressed in an independent manner from any IP network.
- The PAN coordinator acts as a simple IP packet router, so that the traffic generated by any PAN user can be routed to another network, such as a mobile network.

– Provision of an IP-based service is required.

These features can be considered advantages or drawbacks, depending on the requirements of the specific application. For example, the PAN profile can be adequate for applications supported by ad hoc networks, but not for a medical monitoring system based on a body or personal area network, such as the one described in [18]. However, without such conditions, the use of other profiles such as SPP could be preferred. SPP profile, implemented in a great variety of commercial devices, should provide a better performance since the added overhead may be much lower when compared with the equivalent for data transported on TCP/IP (Transport Control Protocol) or UDP/IP (User Datagram Protocol) over BNEP (Bluetooth Network Encapsulation Protocol) [5] under PAN profile.

This study evaluates the effect of the use of these profiles on the performance of a Bluetooth network. The analysis takes into account the overhead added by every protocol involved, as well as the difference between the maximum data units handled in adjacent layers.

This article is structured as follows: Section 2 presents a brief summary of some previous works related to the study of an efficient use of available bandwidth in Bluetooth networks. Most of these works do not take into account the use of Bluetooth profiles. Section 3 then presents a theoretical study to calculate the minimum delay and throughput for both SPP and PAN profiles. Section 4 describes the delay measurements performed in an actual Bluetooth network. The section compares the empirical results with the theoretical values obtained with the proposed formulas. A discussion on the convenience of using each profile follows in Sect. 5. Finally, Section 6 presents the conclusions and the future research lines to continue this work.

2 State of the Art

Quality-of-service (QoS) support is a critical issue to guarantee an optimal usage of bandwidth in mobile applications based on Bluetooth WPANs [18]. Due to the wide range of applications of WPANs, real time data traffic support in a Bluetooth piconet is an issue of great interest in WPAN research. As data traffic is commonly provided as a best-effort service on ACLs (Asynchronous Connectionless Link), there are a significant amount of studies that focus on analysing the efficiency of this type of Bluetooth transmissions [14]. Most of these works investigate the throughput and the end-to-end delay achieved in Bluetooth as a function of the distance between the origin and destination nodes [12], the BER (Bit Error Rate) [13] or the coexistence with 802.11 networks [10,12]. There are many published articles that evaluate the performance of non-IP services. This is the case with [15] and [17]. The polling scheme utilised by the Bluetooth device acting as master is also under research in [19].

There is also a large number of articles that study the performance of TCP on Bluetooth networks, considering different TCP implementations [8,9], segmentation types and polling schemes [7,9]. For example, the study in [8] proposes to fit the Bluetooth packet size and type to the channel conditions so that TCP throughput improves.

In [9] two SAR (Segmentation and Reassembly) policies are studied, SAR Best Fit (BF) and SAR Optimum Slot Utilization (OSU), concluding that the SAR-OSU policy leads to an enhanced performance. Using this method, the same authors investigate the impact of diverse factors, including the buffer size at L2CAP (Logical Link Control and Adaptation Protocol) layer, the selected scheduling algorithm and the TCP version.

The paper in [17] questions the feasibility of many of these proposals aimed at improving the QoS in Bluetooth networks. According to these authors, Bluetooth is a promising wireless technology for Personal Area Networks currently being used for multiple services, based or not on IP. However, the Bluetooth standard does not specify how to comply with quality requirements; although multiple proposals exist, most of these approaches in the literature imply changes in the Bluetooth specification and in existing devices, and, furthermore, they are limited to IP services.

When TCP traffic is analysed, the use of a specific profile for the transport of TCP/IP data is not considered, so IP packets are assumed to be encapsulated in L2CAP PDUs (Protocol Data Units). This is the case of [1] and [9]. In both studies, the objective is to minimise the overhead. These papers show that the overhead is very high for the DUN profile, as it adds overhead of PPP (Point-to-Point Protocol) and RFCOMM protocols to TCP/IP or UDP/IP overhead. In [1] the use of PAN profile is discarded due to the same problem, while in [9] this choice is not considered since that profile was not available at the time of writing the article.

However, the use of profiles cannot be avoided in systems integrating real devices that implement a certain profile, over which the user and even the programmer have little or no control. In this context studies such as [15] and [17] are especially illustrative.

As a consequence of the direct data transmission over L2CAP, studies such as [9] only take into consideration the effect on performance (throughput and end to end delay) of user data fragmentation due to the segmentation at the Baseband layer. On the other hand, the authors in [15] consider the effect of segmentation (segmentation mismatch) that can appear under SPP when there is a mismatch in the maximum data size handled between adjacent layers in the architecture. However, this problem is not considered when the PAN profile, specifically devised for the transport of IP data over Bluetooth, is studied.

The work in [7] proposes some polling schemes orientated towards improving the performance of TCP traffic over Bluetooth. The authors do not just optimize the efficiency of Baseband Bluetooth traffic as other studies do, but they also consider how this traffic is carried over BNEP, setting the MSS (Maximum Segment Size) parameter to a high value to achieve a lower end-to-end packet delay.

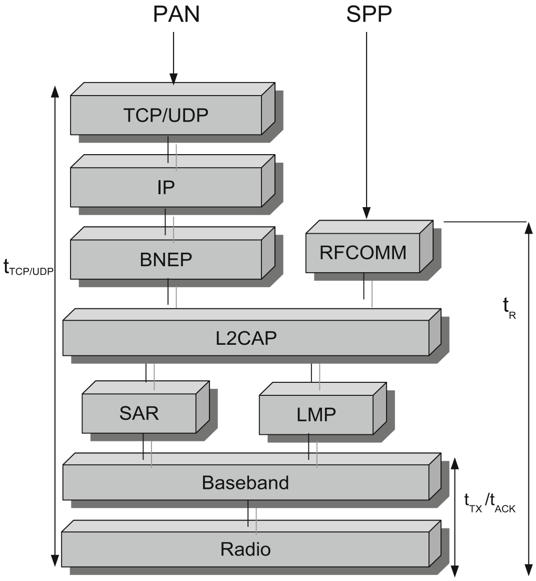
In summary, most existing works ignore the effect of the overhead introduced by the protocols which are required by the particular profiles that a commercial device can implement, as well as the impact of the fragmentation that can occur when there is a mismatch between the maximum transfer units of adjacent layers. The objective of the present article is to evaluate the performance of a Bluetooth personal area network, in which the piconet master acts as coordinator, for the PAN and SPP profiles. The performance will be analytically and empirically assessed in terms of end-to-end delay and throughput, taking into account the impact on delay of the overhead across the Bluetooth architecture and the effect of segmentation mismatch.

3 Theoretical Study of Performance

3.1 End to End Delay for the Optimum Case

This section calculates the minimum transmission delay of Bluetooth ACL transmissions when ideal conditions are assumed, that is, when no retransmission occurs (the Bit Error Rate or BER is null) and the time of storage in the buffers is negligible. The analysis considers the overhead added by all the protocols involved in the transmission when PAN or SPP profile is utilised.

Fig. 1 Bluetooth architecture



The delay for transmitting *N* user bytes is minimum when the information flows from the master to the slave, and assuming that the acknowledgements at the Baseband layer of all the Bluetooth packets are performed in the next Bluetooth slot and with just one slot packet.

The calculation of the delay must take into account the number of bytes in the headers introduced by all the layers, as well as the possible need for fragmentation in any layer to avoid exceeding the MTU of the immediately inferior layer.

Figure 1 shows the transmission delay to be calculated for each profile across the Bluetooth protocol architecture: t_R in the case of SPP and t_{UDP} in the case of PAN. The definition of these delays are detailed in Subsects. 3.1.1 and 3.1.2, respectively. Independent of the profile that is employed, all the information sent by L2CAP is segmented into a series of fragments that are encapsulated in the body payload of the Baseband packets. In order to include this segmentation in the calculation of the transmission delay, we define two components to estimate the delay of transmitting and acknowledging these Bluetooth packets at the Baseband layer. These delay components are:

- $t_{ACK}(N)$: this component describes the time employed to send and confirm every Baseband packet needed to carry *N* octets from the upper layer (L2CAP). Specifically, if *N* exceeds the capacity of a 5-slot packet, the L2CAP PDU will have to be divided in a series of 5-slot packets and a final 1, 3 or 5-slot packet. So, t_{ACK} is computed by the following recursive formula:

$$t_{ACK}(N) = \begin{cases} 0 & N = 0 \\ 2 \cdot T_S & 0 < N \leq L_1 \\ 4 \cdot T_S & L_1 < N \leq L_3 \\ 6 \cdot T_S & L_3 < N \leq L_5 \\ 6 \cdot T_S \cdot \left\lfloor \frac{N}{L_5} \right\rfloor & N > L_5 \\ +t_{ACK}(N \, mod \, L_5) & \end{cases} \qquad (1)$$

In the above formula, T_S represents the duration of a Bluetooth slot (625 µs), while L_1, L_3 and L_5 are the maximum payload sizes for a Baseband 1, 3 and 5 slot packet, respectively. These sizes are 27, 183 and 339 bytes for DH (Data High—Rate) packets and 17, 121 and 224 bytes for DM (Data Medium—Rate) packets.

- $t_{TX}(N)$: this component represents the time required to send N octets from the upper layer without considering the confirmation of the last Baseband packet that bears the final fragment from the upper layer, but taking into account the confirmation of the intermediate fragments. Specifically, when N does not exceed the capacity of a 5-slot packet, this time t_{TX} is calculated as a function of the number of transmitted bits, otherwise, it is computed by invoking the previous equation of t_{ACK} in the following way:

$$t_{TX}(N) = \begin{cases} 0 & N = 0 \\ N_B(N) \cdot T_B & 0 < N \leq L_5 \\ t_{ACK}\,(L_5) \cdot \lfloor N/L_5 \rfloor & \\ +t_{TX}\,(N\ mod\ L_5) & N > L_5 \end{cases} \qquad (2)$$

where T_B is the transmission time for 1 bit (1 µs) and N_B is the number of bits contained in the last Bluetooth packet. This number can be computed as $N_B(N) = N_{pl}(N) + N_{ov}$, being:

- N_{ov}: the 126 bits of control information in the Bluetooth packet (including the packet header of 54 bits and the access code of 72 bits).
- N_{pl}: is the number of bits in the Bluetooth payload, computable as:

$$N_{pl}(N) = \begin{cases} (N + H_S + H_{CRC}) \cdot 8 & DH\ packets \\ \left\lceil \frac{[(N+H_S+H_{CRC})\cdot 8]}{10} \right\rceil \cdot 15 & DM\ packets \end{cases} \qquad (3)$$

where H_{CRC} corresponds to the 2 bytes of the CRC (Cyclic Redundancy Check) field while H_S is the payload header ($H_S = 1$ byte for 1 slot and $H_S = 2$ for 3 and 5 slot-packets, respectively). The previous equation takes into account that for DM packets, which are protected with FEC 2/3 (Forward Error Correction), for every ten information bits five redundancy bits are added. Therefore, if the number of bits is not a multiple of 10, the packet must be completed with extra bits after the CRC field.

3.1.1 Serial Port Profile (SPP)

The RFCOMM protocol is a subset of the ETSI TS 07.10 standard [11] used by the SPP profile to allow the emulation of serial ports communications over L2CAP.

The structure of the frames used in RFCOMM (shown in Fig. 2) is a modified version of that defined in ETSI TS 07.10, excluding the Flag field at the start and the end of the frame. The encoding of the Control field indicating the frame to use is specified in the table 2 of the specification [11].

The frame on which RFCOMM transmits user data is the UIH (Unnumbered Information with Header check) frame. This frame can carry a number of bytes in the information field

Fig. 2 RFCOMM frame structure for basic operation

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

determined by the Maximum Frame Size parameter (*N1*), whose default value is 127 bytes. Even so, this value can be negotiated between 23 and 32767 using the command/response pair PN (Parameter Negotiation).

In the case of UIH frames with the P/F bit (within the Control field) set to zero, the maximum size for the information field is the negotiated value, *N1*, whereas in the case of a credit-based flow control (P/F bit set to one), the structure of the UIH frame is modified, so that the maximum number of credits, *N1*, is decremented in one unit to compensate for the credit field (thus keeping the L2CAP payload size constant).

Since the current analysis does not take this modification into account, the transmission delay, t_R, to transmit N user data bytes user SPP is described by equation:

$$\begin{aligned} t_R(N) = {} & N_{frag}(N) \cdot t_{ACK}\left(X + O_R(X) + H_{L2CAP}\right) \\ & + t_{TX}\left(L_{rem}(N) + O_R(L_{rem}) + H_{L2CAP}\right) \end{aligned} \tag{4}$$

being:

- X: the size at which RFCOMM must fragment the user data. This value has to guarantee that the L2CAP MTU for RFCOMM, M_R, is not exceeded. Therefore, if the sum of *N1* and the maximum overhead introduced by RFCOMM (5 bytes) exceeds the value of M_R, the user data will be fragmented taking the limit imposed by M_R, instead of using the *N1* parameter as negotiated with the corresponding layer.[1] Thus, the size at which the data will be encapsulated in L2CAP frames can be determined as follows:

$$X = min\left(N1, M_R - 5\right) \tag{5}$$

- $O_R(L)$: number of bytes for the control information of the RFCOMM protocol, 4 or 5 bytes depending on whether the Length Indicator in the RFCOMM header, as shown in Fig. 2, consists of 1 or 2 bytes:

$$O_R(L) = \begin{cases} 4 & 0 < L \leq 127 \\ 5 & L > 127 \end{cases} \tag{6}$$

- H_{L2CAP}: size of L2CAP protocol header (4 bytes).
- $N_{frag}(N) = \lceil \frac{N}{X} \rceil - 1$: number of RFCOMM non-final frames.
- $L_{rem}(N) = ((N-1) \ mod \ X) + 1$: number of bytes of the last RFCOMM frame.

Note that the Bluetooth specification does not recommend a minimum value for M_R (that is to say a minimum L2CAP MTU for RFCOMM), in contrast with the case of BNEP for PAN profile. Consequentely, if *N1* parameter exceeds the L2CAP MTU value (M_R), RFCOMM will have to fragment the user data to fit the L2CAP MTU. Thus, the overhead will be greater than in the case that *N1* parameter is set according to L2CAP MTU. For example, if M_R and *N1* are set to 133 and 1,019 bytes, respectively, the transmission of 1,500 bytes of user data requires 12 RFCOMM frames (11 frames with 128 bytes and an additional frame with 92 bytes) which entails an overhead of 107 bytes $(11 \cdot (O_R(128) + H_{L2CAP}) + 1 \cdot (O_R(92) + H_{L2CAP}))$. However, if M_R is set to 1,024 bytes, only 2 RFCOMM frames (with 1,019 and 481 bytes) are necessary, so the overhead is reduced to 18 bytes (9 bytes for each frame).

[1] "It is the responsibility of the higher layer protocol to limit the size of packets sent to the L2CAP layer below the MTU limit" (Part D, Specification of the Bluetooth System [3]).

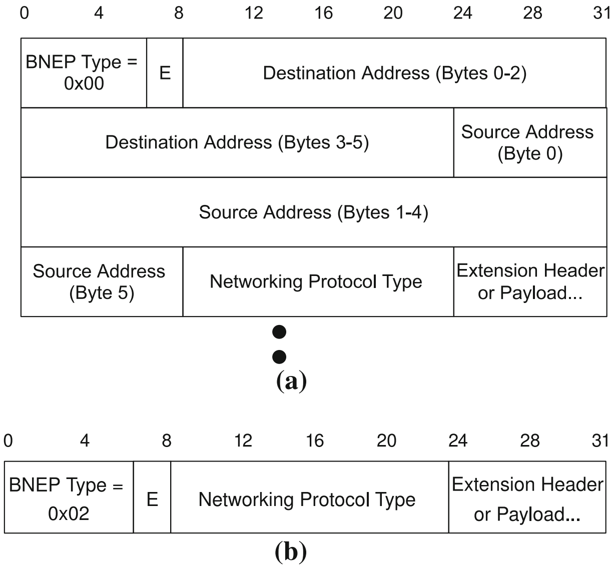


Fig. 3 BNEP Header formats. (**a**) BNEP general Ethernet packet type header, and (**b**) BNEP compressed Ethernet packet type header

3.1.2 Personal Area Network (PAN) Profile

The PAN profile allows the transport of TCP/IP or UDP/IP data over L2CAP using BNEP (Bluetooth Networking Encapsulation Protocol). BNEP replaces the typical layer-2 Ethernet header with a specific header, so that the BNEP header and the Ethernet payload are encapsulated in an L2CAP data PDU. The information in the Ethernet frame is appended with a total of 15 bytes (the BNEP header) if a BNEP General Ethernet packet type (whose structure is shown in Fig. 3a) is used, or a total of 3 bytes in the case of employing a BNEP Compressed Ethernet packet type (described in Fig. 3b). The latter format is utilised when the final destination for the Ethernet data is also a BNEP device. Thus, the compressed format is used when the origin and the destination of an Ethernet frame correspond to a master-slave pair in a Bluetooth piconet.

Since the Ethernet frame encapsulates an IP datagram, carrying in turn a UDP or TCP datagram, the transmission delay to be calculated will be equal to the transmission delay at the transport layer.

Assuming that UDP is employed, the global delay at the transport layer, $t_{UDP}(N)$, is defined by Eq. 7, expressed as a function of the transmission delay at the IP layer $t_{IP}(N)$ and the 8 bytes corresponding to the UDP header:

$$t_{UDP}(N) = t_{\mathrm{IP}}(N+8) \tag{7}$$

The Eq. 8 calculates the value of $t_{IP}(N)$. This formula takes into account the fragmentation that can occur at the IP layer when the BNEP MTU (M'_B) is exceeded. This MTU is set to the maximum payload of an Ethernet frame, that is, 1,500 bytes. This maximum payload that BNEP accepts from the higher layer, (IP) is equal to the MTU of L2CAP for BNEP (M_B), 1,691 bytes, minus 191 bytes (reserved for the BNEP header plus a possible extension header of up to 176 bytes).

$$t_{IP}(N) = N_{frag}(N) \cdot t_{ACK}(M'_B + H_B + H_{L2CAP}) \\ + t_{TX}\left(L_{rem}(N) + H_{IP} + H_B + H_{L2CAP}\right) \quad (8)$$

being:

- H_{IP}: the number of bytes in the IPv4 header (20 bytes).
- H_B: the number of bytes in the header added by the BNEP protocol. This can be calculated, taking into account the header type, in the following way:

$$H_B = \begin{cases} 15 & BNEP\,Type = 0x00 \\ 3 & BNEP\,Type = 0x02 \end{cases} \quad (9)$$

- H_{L2CAP}: the L2CAP protocol header (4 bytes).
- $N_{frag}(N) = \left\lceil \frac{N}{M'_B - H_{IP}} \right\rceil - 1$: the number of non-final IP datagrams.
- $L_{rem}(N) = \left((N-1)\ mod\ \left(M'_B - H_{IP}\right)\right) + 1$: number of bytes of the last IP datagram.

As it can be observed from the previous formulas, the payload from the transport layer (here represented as N bytes) is fragmented in blocks of a maximum size of M'_B minus the size of the IP header.

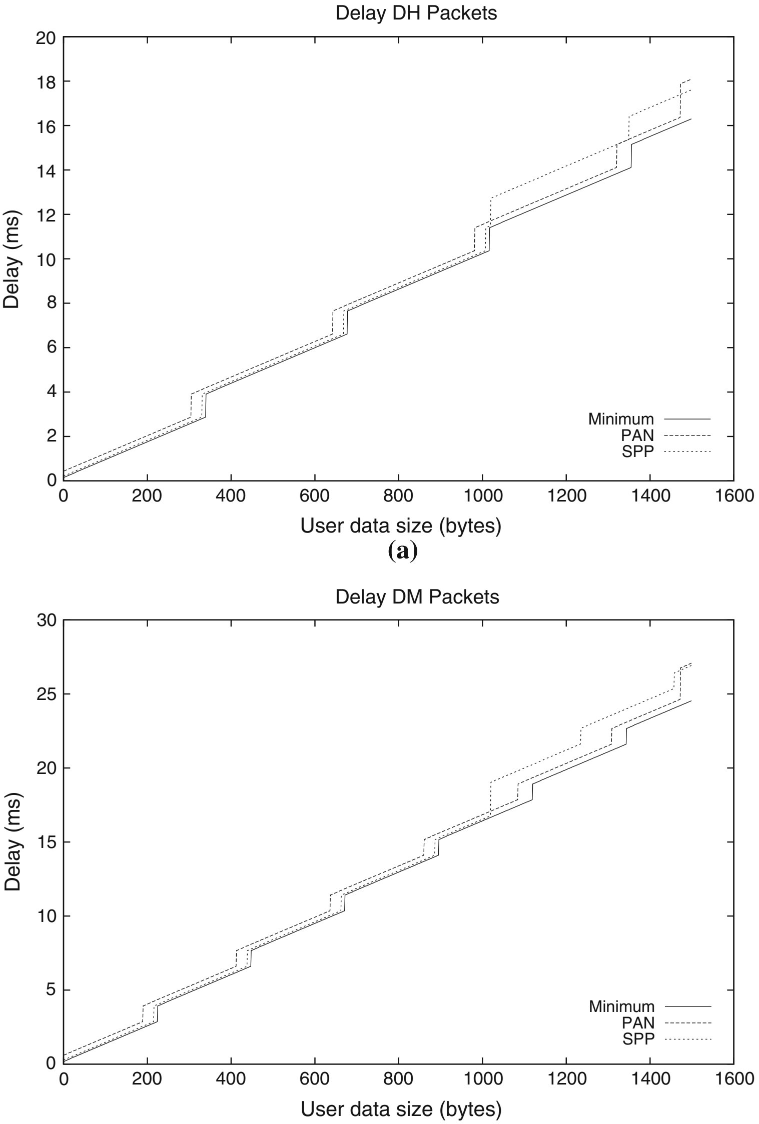
3.2 Evaluation of the Theoretical Results

In this section we show the application of the previous analytical model of the delay for different sizes of the user data. The model has been applied assuming that the L2CAP MTU for RFCOMM (M_R) and BNEP (M_B) are set to 1,024 bytes and 1,691 bytes, respectively. These values are considered as they are the default values defined by BlueZ [6] Bluetooth implementation, which will be utilised for the empirical validation of the model in the next section. In the case of RFCOMM, BlueZ also sets the *N1* parameter to 1,019 bytes, as an implementation choice that avoids the segmentation mismatch effect.

Figure 4 also shows the minimum theoretical delay of the model, $t_{TX}(N)$, obtained from the Eq. 2, for the transmission of user data ranging from 10 to 1,500 bytes, performed by a master device, without using profiles, that is to say, when BNEP or RFCOMM are not present. Additionally, the figure compares this minimum delay with the theoretical ones, $t_R(N)$ and $t_{UDP}(N)$, calculated respectively by Eqs. 4 and 7 for SPP and PAN profiles.

The equally spaced steps appearing in the figures coincide with the filling of 5-slot packets. This is due to the fact that, when the transmission of an L2CAP payload requires more than one 5-slot packet, an acknowledgement must be received before starting the transmission of the next Bluetooth packet. Note that, in the case of SPP, the steps are almost indistinguishable from the ideal case (without profiles), since the overhead added by SPP and L2CAP is insignificant. However, in the case of the PAN profile the steps appear earlier, for sizes of $339 - 35 = 304$ bytes and their multiples, due to the overhead added by UDP (8 bytes), IP (20 bytes), BNEP (3 bytes), L2CAP (4 bytes).

Furthermore, it can be seen how the transmission delay for SPP exceeds that for PAN from data sizes higher than 1,019 bytes, as these sizes surpass the L2CAP MTU specified in the BlueZ RFCOMM implementation. Nevertheless, it must be noted that segmentation mismatch does not take place [15], as it would have in the case that M_R had not been set to a value of $N1 + 5$.



(a)

(b)

Fig. 4 Theoretical delay UDP-PAN versus SPP. (**a**) DH packets, and (**b**) DM packets

To conclude, in order to avoid the segmentation mismatch effect, the L2CAP MTU must be set to the value negotiated with the peer RFCOMM entity, plus the RFCOMM header length.

In the case of BNEP, the figures show that, for user data lengths greater than 1,472 bytes, the delay reflects the fragmentation that takes place at IP layer since the 1,500 bytes (1,472 user data bytes plus 28 bytes of IP and UDP headers) corresponding to the BNEP MTU are surpassed.

Table 1 Maximum theoretical throughput for 5-slot packets

ACL packet type	Size (bytes)	Maximum size (bits)	Thr_{MAX} (kb/s)
DM5	0–224	1,792	477.8
DH5	0–339	2,712	723.2

3.3 Throughput

The maximum theoretical throughput attainable in the transmission of N bytes of user data with a certain Baseband packet, excluding the overhead added to the transmitted bits,can be computed as:

$$Thr_{MAX}(N) = \frac{N_U(N)}{(N_S + 1) \cdot T_S} \tag{10}$$

where N_S is the number of slots filled by the packet (1, 3 or 5 slots), and $N_U(N) = 8 \cdot N$ is the number of useful bits corresponding to N user data bytes, that is, excluding FEC, CRC and payload header. The Table 1 shows how maximum theoretical attainable throughput is calculated according to Eq. 10 for 5-slot packets.

If the overhead added by each layer is included and the user data plus the overhead exceeds the capacity of a 5-slot packet, the real throughput can be calculated according to the equation below:

$$Thr(N) = \begin{cases} \frac{N_U(N)}{(N_{slots}+2)\cdot T_S} & (N_{slots}\, mod\, 2) = 0 \\ \frac{N_U(N)}{(N_{slots}+1)\cdot T_S} & (N_{slots}\, mod\, 2) \neq 0 \end{cases} \tag{11}$$

where N_{slots} is the number of slots that the transmission of the $N_U(N)$ bits requires, taking into account the overhead:

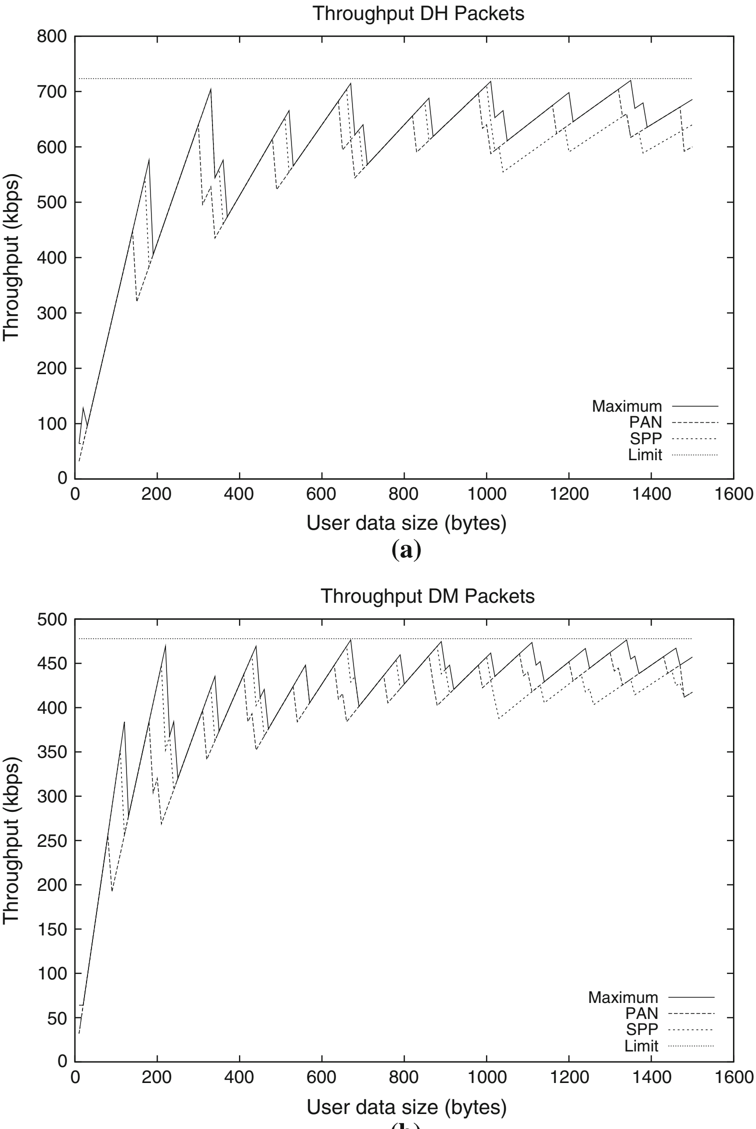
$$N_{slots} = \left\lceil \frac{t_{TX-TOTAL}(N)}{T_S} + 0.4144 \right\rceil \tag{12}$$

In the above expression the transmission time $t_{TX-TOTAL}(N)$ will be computed with Eqs. 2, 4 and 7 to calculate the maximum, SPP and PAN throughputs, respectively. The term $t_{TX-TOTAL}(N)$ is divided by the duration of a slot, T_S (625 µs), to calculate the number of slots. Furthermore, a correction term is added to account for the fact that packets do not fill the last slot because there is a mandatory guard time of 259 µs [3]. This idle guard time represents the 41.44% (259/625) of the duration of the final slot. Finally, the number of slots is obtained by rounding up the resulting value.

It should be remarked that the Eq. 11 discriminates whether the number of slots is even or not. Two slots actually correspond to three slots, and four slots must be considered as five, since transmissions require one, three or five slots. Finally, the acknowledgement slot must be added to the total transmission time, which is a multiple of T_S.

Figure 5 shows the real throughput obtained for SPP and PAN profile as a result of applying the Eq. 11 and implementing the Eq. 12 for $t_R(N)$ and $t_{UDP}(N)$. In this figure the graphs labelled as *Maximum* represent the throughput calculated considering that $t_{TX-TOTAL}(N)$ is t_{TX} (Eq. 2), i.e., neglecting the overhead added by the use of profiles. Finally, the horizontal lines, labelled as *Limit*, corresponds to 477.8 kb/s and 723.2 kb/s, the maximum theoretical attainable throughputs which are calculated by Eq. 10 for DH5 and DM5 packets.

In the case of SPP, it should be clarified that the segmentation mismatch effect does not only occur when the expression $M_R = N1 + 5$ does not hold. This effect also takes place



(a)

(b)

Fig. 5 Theoretical throughput. (**a**) DH packets, and (**b**) DM packets

when the Bluetooth application which uses SPP profile does not configure the *N1* parameter with a value greater than the maximum size of the user data to be transmitted. To highlight the importance of this implementation decision, Fig. 6 shows, for several values of *N1* parameter, how throughput decreases. For all graphs, the M_R has been set according the inequation $M_R \geq N1 + 5$. The curves may be analysed from two perspectives: (a) globally: the optimum case takes place when $N1$ is set to 1,500 bytes (maximum size of user data); (b) for a particular value of *N1*, graphs show that small variations of user data size may reduce the throughput up to a 40%.

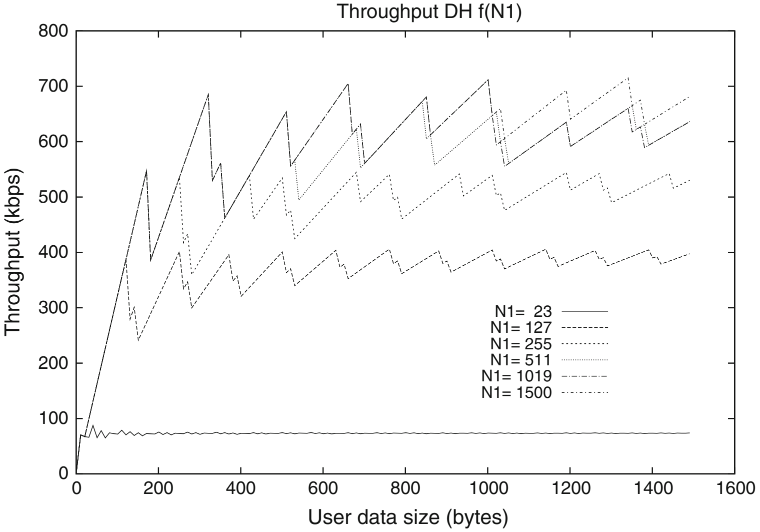


Fig. 6 Example of segmentation mismatch with DH packets

4 Experimental Results

4.1 Measurement Testbed

To evaluate the previous theoretical expressions, different experiments were carried out on real Bluetooth connections employing SPP and PAN profiles.

The measurements of the delay for SPP and PAN have been performed on the same PC with two Bluetooth USB interfaces to avoid synchronization problems. With this objective, separate Client-Server architectures for PAN and SPP have been utilised, both programmed in C on the popular BlueZ protocol stack [6]. To optimise the transmission conditions and minimise any possible interference, both Bluetooth modules were located in a small metal-covered box.

For both profiles, the time elapsed from the start of the packet transmission to its reception has been measured in the destination side. In order to obtain this value, the packet carries the timestamp of its creation. Measurements have been performed in the following situations: (1) master transmitting DH packets, (2) master transmitting DM packets, (3) slave transmitting DH packets and (4) master transmitting DM packets (in order to enforce the use of DH or DM packets, the type of packet employed is configured before each test).

For SPP, the measurement procedure consists in running a server and a client. The server is programmed with the following variable parameters: the mode of operation (transmission or reception), the range of user data sizes to transmit and the interval for the transmissions. So, when an RFCOMM connection is established, the server sends the configuration to the client and then begins to transmit or receive (as appropriate) once the configuration is acknowledged.

In the case of PAN, both the server and client execute a script to establish a PAN network and assign network interfaces. A TCP connection is used to configure the client from the server with the transmission period and the size of the UDP frames to send. Once the testbed is configured, each experiment consists in the transmission through a UDP socket of a user data block of a pre-determined size.

4.2 Comparison of the Theoretical and Measured Delays

This section compares the minimum theoretical delay values computed in the case where the master performs the transmission, and the measured delays for PAN (Fig. 7) and SPP (Fig. 8) in the cases where the master or the slave performs the transmission. For the real Bluetooth transmissions, each point in the figures represents the mean value of 1,000 different transmissions executed with the same user data size (ranging from 1 to 1,500 bytes). The delay for each data block was computed as the time elapsed from the start of the data transmission to the reception of the last data bit in the slave. The delay was measured at both the application and HCI (Host Controller Interface, by means of HCIdump tool [6]) layers. Differences between these two measurement points were always found to be below 1 ms, which indicates that factors such as the Operating System and USB interfaces have a minor impact on the results.

From the figures we can see that the rising slope for the measured delays is the same as that for the theoretical minima, and the steps in the delay curves appear at the same lengths for both the theoretical and the experimental graphs. The constant distance between the theoretical curves and the empirical values is only due to the polling mechanism that coordinates the start of the communication between a Bluetooth master and a slave. As the default poll interval is 25 ms, the delay of each transmission in the empirical testbed is increased (with respect to the theoretical case) between 0 and 25 ms. Accordingly, the polling mechanism introduces a mean offset in the delay of about 12.5 ms.

In the figures we can also note that the graphs of PAN show the increase described in Sect. 3.2, when the number of user bytes is greater than 1,472 bytes.

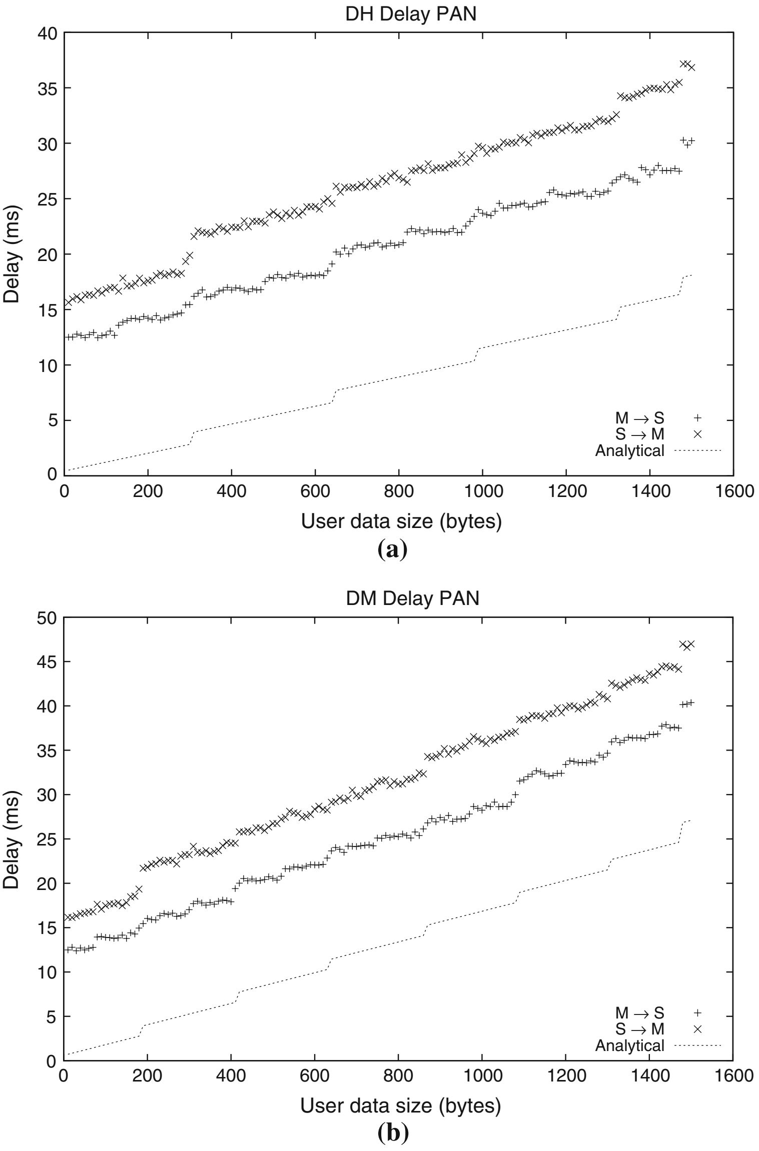
5 Discussion

Although the use of the PAN or SPP profile will be determined by the particular features of the Bluetooth applications to be developed, the transmission of user data using the PAN profile is expected to reach a poorer performance due to the overhead added by TCP/IP (or UDP/IP). Another critical factor to consider is the maximum size handled by the BNEP and RFCOMM protocols, 1,500 bytes in BNEP (maximum size for an Ethernet frame) and up to 32,767 bytes in RFCOMM. However, this configuration is not significant if the relationship with the L2CAP MTU is disregarded.

In fact, it is worth noting the loss of performance that takes place when the MTU for each protocol layer and the relationship between adjacent layers are ignored.

In the case of PAN, the BNEP specifications set the values of the MTU of BNEP and L2CAP to guarantee that no segmentation mismatch effect will happen. However, this is not the case for SPP.

From this study we can conclude that, in order to minimize the segmentation mismatch effect in SPP, the following relation must hold between the *N1* parameter (Maximum Frame Size, negotiated between peer RFCOMM entities) and the L2CAP MTU: $M_R \geq N1 + 5$. Otherwise, the performance may fall below that of the PAN profile, based on BNEP, which initially shows a lower performance as a consequence of the excessive overhead added by the UDP (or TCP), IP and BNEP protocol headers.



(a)

(b)

Fig. 7 Comparison of the model and measured delay for PAN profile. (**a**) DH packets, and (**b**) DM packets

One important advantage of RFCOMM over BNEP to consider is that the upper limit for the *N1* parameter is 32,767 bytes, so $M_R = N1 + 5$ can be easily accomplished since the maximum payload length for a L2CAP PDU is 65,535 bytes. Therefore, if *N1* and M_R are set to high values, for most practical applications, it would only be necessary to fragment the L2CAP PDU in Baseband-sized packets. However, in the case of BNEP, for user data sizes greater than $1,472(1,500 - 20 - 8)$ bytes, the performance drops since IP fragmentation will occur, adding an overhead of 20 bytes in each segment at this layer, apart from the corresponding BNEP header (up to 15 bytes) and the 4-byte L2CAP header.

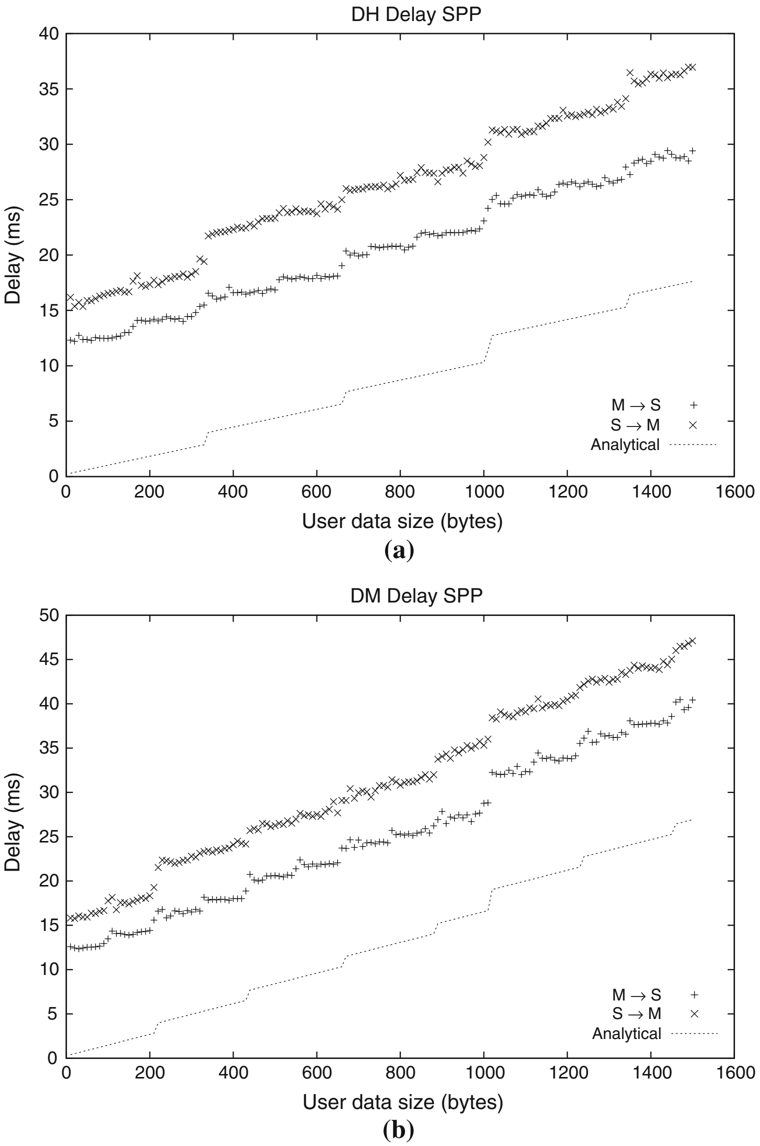


Fig. 8 Comparison of the model and measured delay for SPP profile. (**a**) DH packets, and (**b**) DM packets

To check this limitation of PAN profile, Fig. 9 displays the minimum theoretical delay in the case of SPP, when the L2CAP MTU for RFCOMM M_R and the *N1* parameter are set to a higher value according to the maximum length of the data to transmit, satisfying $M_R = N1 + 5$. Note that in this case RFCOMM outperforms BNEP for all data sizes.

Nevertheless, we have to remark that common programming interfaces for Bluetooth (such as BlueZ) do not normally allow setting the L2CAP MTU value for RFCOMM nor the *N1* parameter. Therefore the segmentation mismatch cannot be avoided.

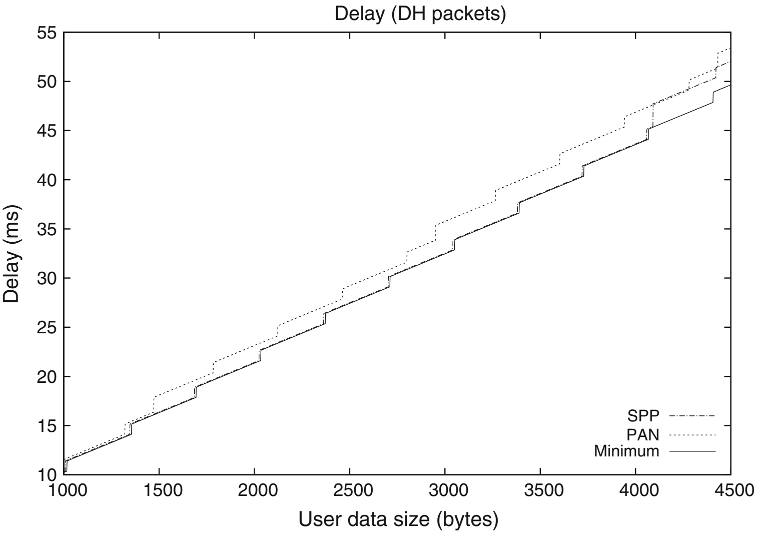


Fig. 9 Theoretical delay UDP-PAN versus SPP – DH packets – $M_R = 4,096$

6 Conclusion and Future Work

The fact that overhead reduction increases the network performance is a trivial and well known conclusion. At present, a huge number of articles about Bluetooth assume that data are sent over L2CAP, or even HCI, to maximize performance. However, as hinted before, this is not an option in systems integrating real devices with a certain profile, as the use of profiles ensures interoperability among Bluetooth devices from different manufacturers.

For this reason the present article has focused on analysing the effect of overhead and segmentation mismatch on throughput and end to end delay when two profiles are utilised: SPP, implemented in a wide variety of commercial devices, and PAN, a profile specifically conceived to deploy Personal Area Networks. In particular, this paper has proposed and validated a model to estimate the minimum delay in Bluetooth transmissions that employ these two profiles. In contrast with other studies, the model considers the overhead and the segmentation introduced at the different layers (RFCOMM, BNEP, L2CAP, Baseband) in the whole Bluetooth protocol stack.

From the performed tests we can conclude that when the application requirements do not mandate the use of the PAN profile, it is recommended to evaluate the use of the SPP profile instead of PAN, since SPP can achieve better performance as the added overhead is substantially lower than that added for TCP/IP transport over BNEP.

In order to minimize the segmentation mismatch effect that takes place in the case of SPP, and, therefore, improve the throughput, it is enough to meet two requirements:

- Negotiation of a value for the *N1* parameter nearest to the maximum length of data to transmit, as *N1* determines the maximum number of user information bytes susceptible of being transported on RFCOMM frames.
- Selection of a value for the L2CAP MTU equal to the *N1* parameter value plus the 5 bytes of overhead of the RFCOMM protocol.

In the case of the PAN profile, since the BNEP protocol (on which it is based) sets the L2CAP MTU to 1,691 bytes and the size of the maximum BNEP data unit to 1,500 bytes, the criterion that avoids decrease in the performance is limiting the user data length to 1,472 bytes (in the case of UDP).

Finally, this paper is presently being extended to study the performance of PAN and SPP profiles in the following cases:

- The addition of more devices to a personal area network, considering both *piconet* and *scatternet* networks.
- The provision of the functionality of access point to other networks (Ethernet, 802.11 and GPRS) in the PAN coordinator.
- The existence of losses in the transmission.

References

1. Bell, D., Cano, J.-C., & Manzoni, P. (2003). Evaluating Bluetooth performance as the support for context-aware applications. In *Proceedings of the 12th International Confernce on Computer Communications and Networks*, 2003 (pp. 345–350).
2. Bluetooth-SIG. (2001a). Bluetooth personal area networking profile revision 0.95a.
3. Bluetooth-SIG. (2001b). Specification of the Bluetooth system, Vol. 1: Core. Bluetooth SIG.
4. Bluetooth-SIG. (2001c). Specification of the Bluetooth system, Vol. 2: Profiles. Bluetooth SIG.
5. Bluetooth-SIG. (2003). Bluetooth network encapsulation protocol (BNEP) specification revision 1.0.
6. BlueZ-Project. (2006). BlueZ protocol stack. http://www.bluez.org.
7. Chan, K., Misic, V., & Misic, J. (2004). Efficient polling schemes for Bluetooth picocells revisited. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, 2004 (pp. 307–314).
8. Chen, L.-J., Kapoor, R., Sanadidi, M., & Gerla, M. (2004). Enhancing Bluetooth TCP throughput via link layer packet adaptation. In *Proceedings of the IEEE International Conference on Communications*, 2004 (Vol. 7, pp. 4012–4016).
9. Das, A., Ghose, A., Razdan, A., Saran, H., & Shorey, R. (2001). Enhancing performance of asynchronous data traffic over the Bluetooth wireless ad-hoc network. In *Proceedings of the IEEE 20th Annual Joint Conference of the IEEE Computer and Communications Societies* (Vol. 1, pp. 591–600).
10. de Cordeiro, M. C., Abhyankar, S., Toshiwal, R., & Agrawal, D. P. (2004). BlueStar: enabling efficient integration between Bluetooth WPANs and IEEE 802.11 WLANs. *Mobile Networks and Applications, 9*(4), 409–422.
11. ETSI. (1997). TS 101 369 (GSM 07.10) V6.3.0. ETSI.
12. Francia, G. A., Kilaru, A., Phuong, L., & Vashi, M. (2004). An empirical study of Bluetooth performance. In *Proceedings of the 2nd annual Conference on Mid-south college computing* (pp. 81–93).
13. Huang, P., & Boucouvalas, A. C. (2005). Delay analysis for Bluetooth baseband ACL packets. In *Proceedings of the Convergence of Telecommunications, Networking & Broadcasting Symposium*, 2005 (pp. 396–401).
14. Ma, M., & Low, S. Y. (2005). Supporting real-time service in Bluetooth networks. In *Proceedings of the Workshop on High Performance Switching and Routing*, 2005 (pp. 167–171).
15. Okura, H., Kato, M., & Tasaka, S. (2002). The effect of segmentation mismatch on quality of continuous media transmission by Bluetooth. In *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2002 (Vol. 2, pp. 702–709).
16. Shinde, H., & Borse, M. (2005). High-rate wireless personal area networks [multimedia capable]. In *IEEE International Conference on Personal Wireless Communications*, 2005 (pp. 19–23).
17. Wu, T.-H., Ke, C., Shieh, C., & Hwang, W. (2004). A practical approach for providing QoS in Bluetooth Piconet. In *Proceedings of the Med-Hoc-Net, 2004*.
18. Yao, J., Schmitz, R., & Warren, S. (2005). A wearable point-of-care system for home use that incorporates plug-and-play and wireless standards. *IEEE Transactions on Information Technology in Biomedicine, 9*(3), 363–371.
19. Yen, L.-H., & Liao, C.-H. (2005). Round-robin with FCFS preemption: a simple MAC scheduling scheme for Bluetooth piconet. In *Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, 2005 (Vol. 1, pp. 761–766).
20. Záruba, G. V., & Johansson, P. (2004). Guest editorial: advances in research of wireless personal area networking and Bluetooth enabled networks. *Mobile Networks and Applications, 9*(1), 7–8.