

Characterization of Bluetooth Packet Delay in Noisy Environments

M. J. Morón, R. Luque, E. Casilari, and A. Díaz-Estrella

Abstract—This letter presents an analytical model to compute the delay of Bluetooth transmissions with Serial Port Profile (SPP) when interferences cause packet retransmissions. In particular, the model takes into consideration the overhead and segmentation introduced by the protocol stack as well as the extra delay provoked by the retransmissions. The model is empirically validated through the measurements of actual Bluetooth connections.

Index Terms—Bluetooth, personal communication networks, serial port profile, protocols.

I. INTRODUCTION

SERIAL Port Profile (SPP) is by far the most implemented profile in commercial Bluetooth (BT) data devices. An adequate characterization of the performance of BT connections with SPP is of great interest to evaluate the practical applicability of this technology. In this sense, BT communications may be seriously affected by the interferences of other wireless systems operating in the same unlicensed 2.4 GHz band. In particular, interfering devices may induce unrecoverable errors in the BT packets and, consequently, the retransmission of the packets.

There are significant research works that analyze the performance of Bluetooth piconets in the presence of interfering sources [1], [2]. Some of these studies empirically characterize the behavior of real BT networks, without providing any analytical model. Conversely, proposed models of BT performance are rarely experimentally validated. In any case, these studies neglect the data segmentation at the different layers of the BT protocol stack for the particular BT profile that is being employed.

The authors in [3] presented an analytical model that characterizes the end-to-end data delay of BT connections with SPP in ideal conditions (with no packet retransmission). This letter extends this model to offer an analytical expression that permits to compute the mean BT packet delay under noisy conditions.

II. MODEL FOR SPP DELAY WITH PACKET RETRANSMISSIONS

The goal is to analytically describe the relationship between the mean BT packet delay and the noise of the environment, described by the mean Bit Error Rate (*BER*) of the transmissions. Assuming that bit errors occur in an uncorrelated way, the *BER* defines the probability $p(N)$ that a BT packet of N

data bytes has to be retransmitted because of an unrecoverable error. If errors followed a ‘burst’ pattern, erroneous bits would tend to concentrate in the same packets, so the number of retransmissions would most probably decrease. Thus, the consideration of the bit errors as an uncorrelated process can be regarded as a pessimistic case for the packet delay.

If bit losses in the payload are independent of the losses in the header, we can calculate $p(N)$ as:

$$p(N) = p_D(N) + p_H - p_D(N) \cdot p_H \quad (1)$$

where $p_D(N)$ and p_H are the probabilities of retransmitting a packet because of an error in the packet payload or in the packet header, respectively.

The BT packet header consists of 54 bits organized in 18 groups of 3 bits with FEC 1/3 protection (for every information bit, two redundancy bits are added). Therefore, the probability (p_{NEG}) of having less than two errors per group is:

$$p_{NEG} = (1 - BER)^3 + 3 \cdot BER \cdot (1 - BER)^2 \quad (2)$$

Consequently, the probability of experiencing at least one unrecoverable error in the whole packet header (so that the BT packet has to be retransmitted) can be calculated as:

$$p_H = 1 - p_{NEG}^{18} \quad (3)$$

The probability $p_D(N)$ of an unrecoverable error in the data payload depends on the packet type that is being utilized. In DM (Data Medium-Rate) packets, 2/3 FEC protection adds five redundancy bits for each group of 10 data bits so that the algorithm can correct a single bit error in any 15-bit group. The probability (p_{BDM}) of having less than two errors in any group is:

$$p_{BDM} = (1 - BER)^{15} + 15 \cdot BER \cdot (1 - BER)^{14} \quad (4)$$

From this value, $p_D(N)$ can be computed as:

$$p_D(N) = 1 - (p_{BDM})^{n_{gDM}(N)} \quad (5)$$

where $n_{gDM}(N)$ is the number of 15-bit groups in which the payload data are distributed:

$$n_{gDM}(N) = \left\lceil \frac{n_D(N)}{10} \right\rceil \quad (6)$$

being $\lceil y \rceil$ the lowest integer higher than y and $n_D(N)$ the number of bits in the payload, which can be calculated as:

$$n_D(N) = (N + H_P(N) + O_{CRC}) \cdot 8 \quad (7)$$

where O_{CRC} corresponds to the 2 bytes of the Cyclic Redundancy Check overhead while $H_P(N)$ is the number of bytes

This work was supported by National Project No. TEC2006-12211-C02-01. The authors are with the Departamento de Tecnología Electrónica, University of Málaga in Spain (e-mail: {mjmoron, rluque, ecasilari, adiaz}@uma.es).

of the payload header, 1 or 2 bytes depending on the number of slots required by the BT packet:

$$H_p(N) = \begin{cases} 1 & N \leq L_1 \\ 2 & N > L_1 \end{cases} \quad (8)$$

where L_1 is the maximum size of the payload of a 1-slot BT packet (17 bytes for the DM type).

If a 'reliable channel' is assumed, the BT recommendation does not limit the number of retransmissions of the same packet. As a consequence, the Baseband shall continue retransmitting the packet until it is properly acknowledged or a link loss occurs. So, once $p(N)$ is known, we can derive the mean number of times ($\overline{N_{RTx}}(N)$) that a BT packet of N data bytes has to be transmitted:

$$\overline{N_{RTx}}(N) = (1 - p(N)) \cdot \sum_{i=1}^{\infty} i \cdot p(N)^{i-1} = \frac{1}{1 - p(N)} \quad (9)$$

Besides, SPP defines the use of RFCOMM protocol. RFCOMM organizes user data in frames which are sent to the lower layers of BT stack via L2CAP (Logical Link Control & Adaptation Protocol). The data fragmentation at RFCOMM is performed so that there is always just one RFCOMM frame contained in each L2CAP frame. Finally, before being emitted to the radio medium, L2CAP frames are fragmented in a series of BT packets at the Baseband Layer.

Taking into account the fragmentation at the different layers together with the effect of the retransmissions, the mean time for the transmission of N_U user data bytes through SPP can be calculated as:

$$\overline{t_R}(N_U) = \frac{T_{poll}}{2} + n_{nff}(N_U) \cdot \overline{t_{ACK}}(L_R + O_R(L_R) + H_L) + \overline{t_{TX}}(L_{ff}(N_U) + O_R(L_{ff}(N_U)) + H_L) \quad (10)$$

The terms in this equation are defined as follows:

- $T_{poll}/2$: mean delay due to the BT polling scheme, which obliges the master to address the slaves at regular intervals (T_{poll}). So, when data at the application layer are ready to be sent, the transmission may still wait an extra period which can be approximated by a uniform distribution with mean $T_{poll}/2$.
- L_R : Size of the frames into which RFCOMM layer splits the user data. This size is limited by both the *Maximum Frame Size* (N_1) of RFCOMM and the Maximum Transfer Unit (MTU) of L2CAP for RFCOMM (M_R):

$$L_R = \min(N_1, M_R - O_R(N_1)) \quad (11)$$

being $O_R(x)$ the overhead introduced by RFCOMM in each frame: 5 bytes if the data (x) exceed 127 bytes and 4 bytes in other case.

- $n_{nff}(N_U)$: Number of non-final RFCOMM frames in which the N_U user data bytes are fragmented:

$$n_{nff}(N_U) = \left\lceil \frac{N_U}{L_R} \right\rceil - 1 \quad (12)$$

- H_L : Size of the L2CAP header (4 bytes).
- $L_{ff}(N_U)$ is the number of data bytes of the last RFCOMM frame:

$$L_{ff}(N_U) = ((N_U - 1) \bmod L_R) + 1 \quad (13)$$

- $\overline{t_{ACK}}(x)$: Mean delay required to send all intermediate Baseband BT packets in which RFCOMM frames are divided [3]. As every BT packet must be acknowledged in the next slot and with a single slot packet, we have that:

$$\overline{t_{ACK}}(x) = \begin{cases} 0 & x = 0 \\ 2 \cdot T_S + (\overline{N_{RTx}}(x) - 1) \cdot T_{poll} & x \leq L_1 \\ 4 \cdot T_S + (\overline{N_{RTx}}(x) - 1) \cdot T_{poll} & L_1 < x \leq L_3 \\ 6 \cdot T_S + (\overline{N_{RTx}}(x) - 1) \cdot T_{poll} & L_3 < x \leq L_5 \\ (6 \cdot T_S + (\overline{N_{RTx}}(L_5) - 1) \cdot T_{poll}) \left\lfloor \frac{x}{L_5} \right\rfloor + \overline{t_{ACK}} \cdot (x \bmod L_5) & x > L_5 \end{cases} \quad (14)$$

where T_S is the duration of a Bluetooth slot (625 μ s), while L_1 , L_3 and L_5 are the maximum sizes of the payload of a 1, 3 and 5-slot Bluetooth packet, respectively (17, 121 and 224 bytes for DM packets).

- $\overline{t_{TX}}(x)$: Mean transmission time of the final RFCOMM frame. In this case, as the transmission concludes when the last packet of the final frame is received in the BT slave, the final acknowledgement slot has not to be computed:

$$\overline{t_{TX}}(x) = \begin{cases} 0 & x = 0 \\ T_S + (\overline{N_{RTx}}(x) - 1) \cdot T_{poll} & x \leq L_1 \\ 3 \cdot T_S + (\overline{N_{RTx}}(x) - 1) \cdot T_{poll} & L_1 < x \leq L_3 \\ 5 \cdot T_S + (\overline{N_{RTx}}(x) - 1) \cdot T_{poll} & L_3 < x \leq L_5 \\ (6 \cdot T_S + (\overline{N_{RTx}}(L_5) - 1) \cdot T_{poll}) \left\lfloor \frac{x}{L_5} \right\rfloor + \overline{t_{TX}}(x \bmod L_5) & x > L_5 \end{cases} \quad (15)$$

In the equations (14) and (15) the term $(\overline{N_{RTx}}(x) - 1) \cdot T_{poll}$ represents the mean delay introduced by the retransmissions as $(\overline{N_{RTx}}(x) - 1)$ is the mean number of times that a packet (of size x) is retransmitted. Note that, due to BT polling scheme, when an unrecoverable bit error is detected, the packet has to wait a polling interval (T_{poll}) to be retransmitted.

III. EMPIRICAL MODEL EVALUATION

We checked the accuracy of the proposed model by measuring the end-to-end delay in systematic transmissions programmed for an actual BT network of two nodes employing SPP and residing in the same equipment (a PC with two USB Bluetooth adapters). For each considered packet size (between 10 and 1500 bytes) a series of 1000 packets of a predetermined size were sent from the master to the slave. The mean transmission delay for each packet size was computed in the receptor.

In order to induce errors in the bits transmitted by Bluetooth and, consequently, packet retransmissions (in case that the errors cannot be recovered), we located an interfering Wi-Fi traffic source in the vicinity of the BT interfaces. In particular an 802.11g connection (also operating in the same 2.4 GHz ISM band of Bluetooth) was established between the PC with the two BT modules under test and an 802.11g Access Point (also performing as IP router). For this purpose, an

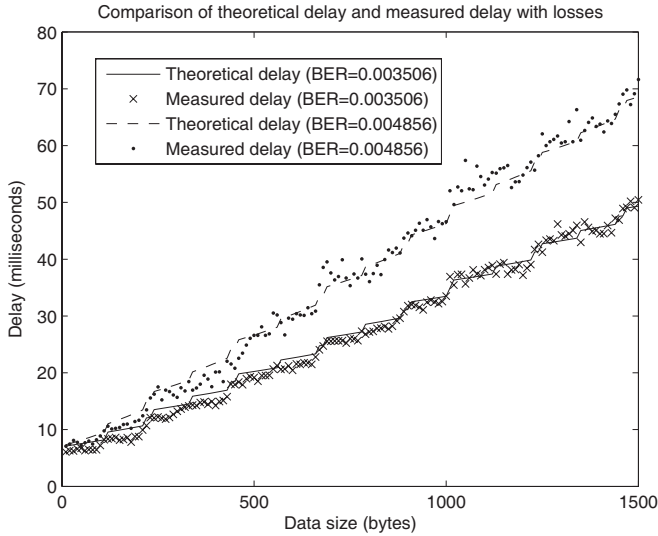


Fig. 1. Comparison of the theoretical delay computed with the model and the measured delay in the actual BT transmissions.

802.11g PCMCIA interface is connected to the PC while the connection is accomplished through an UDP socket between the PC and another terminal connected via Ethernet with the Access Point.

To increase the noise introduced in the BT transmissions the Access Point was situated only 60 cm away from the BT interfaces. Similarly, the tests were executed under heavy traffic conditions in the interfering 802.11g connections, with a bit rate between the second terminal and the PC of up to 10 Mbit/s. Using *iperf* tool [4], this background traffic was generated with a constant bit rate source emitting 1470-byte packets.

In our testbed, the BER of every experiment is indirectly calculated from the Link Quality (LQ) estimation which is provided by the HCI (Host Controller Interface) layer of the CSR Bluetooth Chipset [5] utilized by the BT interfaces. LQ is an integer (discrete) value between 0 and 255. According to CSR chipset specification (see [5], [6] for more details), the BER can be estimated from LQ with the following formula:

$$BER = \begin{cases} (255 - LQ) \cdot 0.25 \cdot 10^{-4} & 215 \leq LQ \leq 255 \\ 0.001 + (214 - LQ) \cdot 8 \cdot 10^{-4} & 90 \leq LQ \leq 215 \\ 0.1 + (89 - LQ) \cdot 64 \cdot 10^{-4} & 0 \leq LQ \leq 90 \end{cases} \quad (16)$$

After the corresponding 1000 transmissions of each packet

size, a command is sent to the BT modules to compute the LQ and the corresponding BER.

Figure 1 depicts the measured mean delays for two different interference conditions (regulated by the traffic load generated in the interfering source). These noise conditions are characterized by the mean of the values of the BER estimated for the different packet sizes. The graphs also show the delay that is computed by the analytical extended model when this mean BER value is considered as an input in the model to account for the effects of the retransmissions. The graphs illustrate that the analytical model can accurately fit the empirical results (a similar performance of the model has been detected for other interference conditions). Thus, the model could be appropriate to predict the performance (and even the usability) of Bluetooth technology for a communication application that has to be deployed in noisy environments.

IV. CONCLUSIONS

This work has proposed an analytical expression to compute the mean packet delay of Bluetooth transmissions with SPP profile under noisy conditions. The model determines the delay as a function of the probability of retransmitting a packet, which is in turn forwardly derived from the Bit Error Rate. The empirical evaluation with actual devices shows the capability of the model to predict the performance of Bluetooth depending on the application data size and the environmental conditions. The model has been conceived for DM packets of Bluetooth 1.0 and 1.1 but it can be easily extended to Bluetooth 2.0 and DH (Data High Rate) packets.

REFERENCES

- [1] T. Khoutaif and G. Juanole, "Formal modelling and evaluation of the data transfer phase of the ACL links on the WPAN Bluetooth," in *Proc. IEEE Conference on Emerging Technologies and Factory Automation*, pp. 30–37, Sept. 2006.
- [2] A.-C. Hsu, D. Wei, and C.-C. Kuo, "Coexistence mechanism using dynamic fragmentation for interference mitigation between Wi-Fi Bluetooth," in *Proc. Military Communications Conference*, pp. 1–7, Oct. 2006.
- [3] M. Moron, R. Luque, E. Casilari, and A. Diaz Estrella, "Minimum delay bound in Bluetooth transmissions with serial port profile," *Electron. Lett.*, vol. 44, no. 18, pp. 1099–1100, 2008.
- [4] Iperf, "Internet Protocol bandwidth measuring tool." URL: <http://iperf.sourceforge.net/>.
- [5] CSR, "Cambridge Silicon Radio Plc. BlueCore Bluetooth chipset." URL: <http://www.csr.com>.
- [6] L.-J. Chen, R. Kapoor, M. Sanadidi, and M. Gerla, "Enhancing Bluetooth TCP throughput via link layer packet adaptation," in *Proc. IEEE International Conference on Communications*, vol. 7, pp. 4012–4016, June 2004.