Mathematical Analysis of Bluetooth Energy Efficiency

Andrea Zanella, Silvano Pupolin

e-mail:

{andrea.zanella, silvano.pupolin}@dei.unipd.it
Abstract

In this paper, we propose a mathematical framework for the analysis of Bluetooth systems. The dynamic of the system is modelled by means of a finite state Markov chain (FSMC). Hence, we resort to the renewal reward theory to derive an estimation of the average throughput and energy efficiency achieved by the different packet formats, for both AWGN and Rician fading radio channels. System behavior is, then, investigated under a wide range of parameters, like receiver–correlator margin, average signal to noise ratio and Rice factor. The analysis we present may provide precious guidelines for the design of energy–efficient Segmentation–and–Reassembly modules and baseband polling algorithms for Bluetooth piconets.

I. INTRODUCTION

Bluetooth ([1], [2], [3], [4]) is an emerging radio technology that is expected to play a leading role, in the near future, in the field of short–range personal communications. Although Bluetooth can hardly compete in terms of transmission speed with other existing radio technologies, like IEEE 802.11b, it is definitely competitive in terms of energy consumption. Bluetooth, indeed, was designed to be integrated in portable, battery driven electronic devices, for which energy saving is a key issue. In order to meet this goal, Bluetooth standard defines an appropriate reception mechanism at the baseband layer. Active units are synchronized to a common time–slotted channel that is periodically scanned for valid packets. The scanning lasts just for the time needed to recognize a valid packet. After this time, units that are not recipient of any packet can switch off their transceivers till the successive scan interval. In this way, a unit that is not addressed by any valid packet is active for no more than 10% of the time.

Although the reception mechanism is well defined by the Bluetooth standard, many aspects related to the energy efficiency achieved by the system still need to be investigated. One of these aspects is related to the performance achieved by the different baseband packet formats provided by the standard, at the varying of radio channel conditions. Indeed, Bluetooth provides six data packet types, which differ for time duration, error protection and data capacity. Unprotected and long packet types show high payload capacity but are sensitive to payload errors. On the contrary, short and protected types are less subject to payload errors to the detriment of a lower capacity. There is, then, a potential performance tradeoff among the different packet formats that deserves to be investigated.

Another interesting issue is the impact on system performance of an important design parameter, namely the receiver–correlator margin $S$. Loosely speaking, this parameter determines the selectivity of the receiver with respect to packets containing errors. Low margin values imply strong selectivity, with the risk of dropping packets that could be successfully recovered. On the contrary, high values imply weak selectivity, with the risk of receiving an entire packet before realizing that it contains unrecoverable errors. Therefore, the receiver–correlator margin $S$ might potentially determine a tradeoff between throughput and energy consumption.

Furthermore, Bluetooth networking is based on a small network structure, called piconet, in which a unit assumes the role of master, while the others act as slaves. Thus, it may be worth investigating the way in which master and slave units drain their energy at the varying of the traffic pattern and channel conditions.

In the following, we investigate these topics by means of a simple mathematical model for the Bluetooth point–to–point connection. The dynamic of the system is captured by means of a Finite–State Markov Chain (FSMC) model. Hence, following the approach suggested in [5], we resort to the renewal reward analysis to compute the average throughput and energy performance achieved by the system. The analysis is carried out in both AWGN and Rician fading radio channel. System behavior is, then, investigated under a wide range of parameters, like packet type used in downlink and uplink connection, receiver–correlator margin, average signal to noise ratio.

The topic we consider in this paper has been partially addressed by other works in the literature (see, for instance, [6–9]). To the authors knowledge, however, the literature still lacks in accurate performance analysis that takes into
consideration, beside delay and throughput, also energy consumption in the specific case of the Bluetooth system. The novelty of the work we present in this paper, hence, lies in the mathematical analysis of the system performance in terms of energy. The analysis considers in details the reception mechanism defined by the Bluetooth standard and the characteristics of each packet type. This approach allows us to derive simple mathematical expressions for the amount of useful data delivered and energy consumed by master and slave units, for all the possible combinations of transmission/reception error events. The results we obtain in terms of energy efficiency and average throughput may provide useful guidelines for the design of energy–aware algorithms for the piconet management.

The remainder of this paper is organized as follows. Section II provides an overview of the Bluetooth radio system. In Section III, we derive the mathematic model used to describe the system dynamic. Section IV presents a detailed performance analysis, based on the provided mathematical model. Finally, Section V provides concluding remarks.

II. BLUETOOTH RADIO SYSTEM

A. Baseband

Bluetooth operates in the Industrial Scientific Medical (ISM) frequency band, centered around 2.4GHz. It uses a binary Gaussian–shaped Frequency Shift Keying (GFSK) modulation, with a bit period of $1 \mu$s, achieving a raw bit rate of 1 Mbit/s. A fast frequency hopping spread spectrum scheme is used to limit interference from and to other radio devices operating in the same band. The frequency band is partitioned in a set of 79 RF channels, 1 MHz wide each. The carrier frequency jumps in this set, on the basis of a pseudo–random hopping sequence, with a nominal hop rate of 1600 hops/s (1 hop every $625\mu$s).

In order to communicate, Bluetooth units have to be organized in a small network, called piconet. A piconet can host up to eight active units, one of which assumes the role of master, while the others become slaves. All the units in the same piconet are time and frequency synchronized to a Frequency Hopping channel. The frequency hopping sequence can be uniquely derived from the Bluetooth address and clock of the master unit.

Transmissions can directly occur between master and slaves only. Duplex communication is obtained by a slot–based Time Division Duplex scheme. Time is divided into consecutive slots that are used for downlink (master-to-slave) and uplink (slave-to-master) transmissions, alternatively. Each slot has a duration of $T_{\text{slot}} = 625\mu$s. In general, the carrier frequency is changed at each time slot. However, the carrier frequency cannot be changed during the transmission of a packet, so that multi-slot packets are transmitted on the same frequency.

In order to prevent collisions among units in a piconet, the master employs a simple polling technique to enable each slave unit to transmit. On the basis of this scheme, only the slave addressed by a downlink packet is allowed (and required) to transmit a packet to the master in the following uplink slot. The master can poll the slave implicitly, by using a useful data packet (if any), or explicitly, with a short control packet (POLL) that does not contain the payload field. The recipient slave is required to reply immediately to the master by transmitting a data packet or a special control packet (NULL) with no payload.

B. Data packet formats & Reception Mechanism

Bluetooth supports both synchronous connection oriented (SCO) and asynchronous connectionless (ACL) links. SCO links are used for voice traffic, while ACL links provide a basic point–to–point connection for asynchronous data traffic. In the following of this paper, we will consider ACL links only.

An ACL packet can extend over an odd number of consecutive slots, namely, one, three or five slots. The transmission of a new packet is preceded by an idle period of $220\mu$s. This period is used by units for processing the previous
packet and synchronizing on the new carrier frequency. Each baseband packet contains three main fields: Access Code (AC), Packet Header (HEAD) and, optionally, Payload (PAYL), as depicted in Fig. 1.

The AC field is used for synchronization and piconet identification. All the packets exchanged within the same piconet have the same AC field. The AC is 72–bit long and contains a synchronization word that assures a minimum Hamming distance of 14 between ACs of different piconets. At the beginning of each receive slot, the Bluetooth receiver correlates the incoming bit stream against the expected synchronization word. For an incoming packet to be recognized, the correlator output has to exceed a given threshold. The receiver–correlator margin, denoted by $S$, is the distance between the maximum correlator output (perfect matching between incoming and expected AC words) and the correlator threshold. Hence, an incoming packet is recognized only whether the gap between actual and maximum correlator outputs falls within the correlator margin $S$. In this case, the HEAD field is also received and decoded; otherwise, reception stops and the units sleeps until the successive receive slot (approximately two slots later). Note that, the value of the correlator margin $S$ is not specified by the standard.

The access code field is followed by the packet header (HEAD). It contains 18 bits, coded with a 1/3 forward error correction code (two–time repetition of every bit), resulting in a total field length of 54 bits. HEAD contains link control information, including packet type, destination address, sequence number and acknowledgment flag (ARQN). HEAD includes also an 8 bit Header Checksum field (HEC) that is used to check the integrity of the HEAD information after decoding. If the HEC test fails, the receiver switches off until the following receive slot (approx two slots later). Otherwise, the unit checks the 3–bit destination address field in HEAD. If the unit is not the intended recipient of the packet, then it switches off the transceiver and sleeps until the first receive slot after the end of the incoming packet. In this case, indeed, the unit can determine the packet type from the appropriate field in the packet header. Hence, it will not wake up before the packet has been completely transmitted by the sender.

In general, AC and HEAD fields are followed by the payload field. However, POLL and NULL packet types contain no payload. Such packets are used whenever a unit is required to send a packet, e.g., for polling a slave or acknowledging the reception of a packet, and there is no data available to the designed destination. Except for POLL and NULL, the other ACL data packet types include a data part (PAYL) that can extend over one, three or five consecutive slots. The PAYL field can optionally be protected by a (15,10) shortened Hamming code, which is able to correct all single errors and detect all double error in each codeword. unprotected packet formats are usually denoted by DH5, DH3 and DH1, for the 5, 3 and 1-slot long types, respectively. Analogously, DM5, DM3 and DM1 are used to denote the corresponding protected formats. The main characteristics of the six different data packet formats provided by Bluetooth are summarized in Tab. I.

![Fig. 1. Bluetooth data packet format](image-url)
C. Automatic Retransmission Query mechanism

Bluetooth aims at providing a reliable ACL service. To this end, the standard defines an Automatic Retransmission Query (ARQ) mechanism at the baseband layer. Each data packet is transmitted and retransmitted until acknowledgement of a successful reception is returned by the destination. The integrity of the received data packet is checked by means of a 16–bit cyclic redundancy code (CRC) that is included in the payload field of each data packet. The acknowledgement information is carried by the ARQN flag in the header of the return packet (piggy backing).

When the destination unit receives a data packet, the CRC field is checked. If the CRC checks, the ARQN flag in the HEAD of the return packet is set to one, denoting a positive acknowledgment (ACK). On the contrary case, the ARQN flag is cleared and the packet is negative acknowledged (NAK).

Negative acknowledgement is assumed by default. Hence, a packet that was correctly received can be retransmitted because the piggy–backing acknowledgement in the return packet failed, i.e., either AC or HEAD fields in the return packet contained unrecoverable errors. In this case, the destination keeps receiving the same payload over and over again. It may be worth noting that, because of the poll mechanism, slave units never retransmit packets that were successfully received by the master. Indeed, a slave unit is allowed to transmit only whether it recognizes a valid poll packet from the master, i.e., both AC and HEAD fields are successfully decoded. Therefore, the slave is also acquainted with the reception status of its previous transmission. Thus, slave retransmissions occur only when needed, i.e., in case of reception of a negative acknowledged (NAK).

On the contrary, the master can miss the return packet from a slave and, thus, retransmit a packet that was successively received. In order for a slave to recognize and filter out duplicate packets, a sequence number field is included in the header of each packet. Consecutive packets received with the same sequence number are recognized as duplicates. The slave receives only AC and HEAD of a duplicate packet, while the PAYL field is ignored. For each duplicate packet received, the slave returns an uplink packet that carries in piggy–back the positive acknowledgement for the original downlink packet.

<table>
<thead>
<tr>
<th>Number of slots</th>
<th>PAYL FEC</th>
<th>PAYL-data length (bit)</th>
<th>Total packet length (bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM1</td>
<td>Yes</td>
<td>136</td>
<td>366</td>
</tr>
<tr>
<td>DH1</td>
<td>No</td>
<td>216</td>
<td>366</td>
</tr>
<tr>
<td>DM3</td>
<td>Yes</td>
<td>968</td>
<td>1626</td>
</tr>
<tr>
<td>DH3</td>
<td>No</td>
<td>1464</td>
<td>1622</td>
</tr>
<tr>
<td>DM5</td>
<td>Yes</td>
<td>1792</td>
<td>2870</td>
</tr>
<tr>
<td>DH5</td>
<td>No</td>
<td>2712</td>
<td>2870</td>
</tr>
</tbody>
</table>

III. MATHEMATICAL MODEL

In this section, we define the mathematical model for a simple Bluetooth connection. For the sake of simplicity, we limit the study to the case of a piconet with only two units: one master and one slave. (The extension of the analysis to the multi–slave case would complicate the exposition without adding any relevant concept.) We assume continuous transmission between master and slave units and infinite retransmission timeout. Furthermore, we suppose units never change the packet types used for data transmission. As usual, we consider a slow flat Rician fading model
for the radio channel and we assume the WSSUS (Wide–Sense Stationary Uncorrelated Scattering) hypothesis, so that different frequency carriers are interested by independent fading processes [10]. By virtue of the frequency hopping mechanism, thus, each packet transmission experiments an independent fading statistic.

Under these hypothesis, the dynamic of the system can be described by means of a sequence of independent identically distributed discrete random variables, which take value in the event space $E$. The occurrence of an event $E_i \in E$ determines the evolution of the system for a given number of successive slots, after which another event can occur independently of the past history of the system. In practice, the system is modelled by means of a very simple finite state Markov Chain (FSMC), with state space $E$, such that, for each $E_i, E_j \in E$, the transition probability from state $E_i$ to state $E_j$ is equal to the steady–state probability of the chain being in state $E_j$.

Following the approach suggested in [5], Bluetooth performance can be investigated by resorting to the classical theory of renewal reward processes [11]. Consider two generic reward functions, $A$ and $B$, associated to the finite state Markov chain (FSMC). Let $A_j$ and $B_j$ be the average reward earned each time the Markov chain enters in state $E_j$. Furthermore, let $A(\tau)$ and $B(\tau)$ be the total reward earned through the system evolution in the interval $[0, \tau]$. Then, from renewal theory [12], we can draw the following result:

$$\lim_{\tau \to \infty} \frac{A(\tau)}{B(\tau)} = \frac{\sum_{E_j \in E} \pi_j A_j}{\sum_{E_j \in E} \pi_j B_j};$$

where $\pi_j$ denotes the steady–state probability of the chain being in state $E_j \in E$. A proper choice of the reward functions will allow us to derive a number of performance indexes. In particular, for each state $E_j \in E$, we consider the following functions:

- average number of successfully delivered data bits, $D_j$;
- amount of consumed energy, $W_j$;
- time delay $T_j$.

In the following we specify the state space $E$ and the system dynamic associated to each state $E_i$. Furthermore, we derive the values of the reward functions associated to each state.

### A. State space & Reward functions

We first introduce some notations before presenting our analysis. As seen in Section II, a packet is successfully received only it satisfies three integrity checks, namely: AC, HEC and CRC. Let $AC_{er}$, $HEC_{er}$ and $CRC_{er}$ denote the check error event for AC, HEC and CRC, respectively. Note that, the reception of each field is subordinated to the good recognition of the previous field. Consequently, $HEC_{er}$ event can occur only if the AC of the incoming packet has been successfully received. Analogously, a $CRC_{er}$ event can occur only if both AC and HEAD fields have been successfully received. Consequently, events $AC_{er}$, $HEC_{er}$ and $CRC_{er}$ are disjoint. We refer to the literature (e.g., [13, 14]) for the expressions of the probability of such error events, which will be largely used in the following analysis.

When either AC or HEAD fields of an incoming packet are not successfully decoded, the packet is said to be not recognized. The recognition error event is denoted by $REC_{er}$, hence we have

$$REC_{er} = \{AC_{er}\} \cup \{HEC_{er}\},$$
where the symbol $\cup$ stands for *inclusive or*. The notation $REC_{ok}$ is used to indicate the complementary event of $REC_{er}$, i.e., the successful reception of both AC and HEAD fields. Furthermore, the successful reception of the entire packet is denoted by $PR_{ok}$.

Let $w_{TX}(X)$ and $w_{RX}(X)$ be the amount of energy consumed by a unit for transmitting and receiving, respectively, the generic packet field $X$. Hence, denoting by $P_{TX}$ and $P_{RX}$ the amount of energy per bit consumed by radio transceivers during data transmission and reception, we have:

$$
w_{TX}(X) = P_{TX} \cdot \text{Length}(X);
$$

$$
w_{RX}(X) = P_{RX} \cdot \text{Length}(X);
$$

where $\text{Length}(X)$ is the total number of bits in the packet field $X$. Furthermore, let $w_0$ be the *average* amount of energy consumed by the receiving unit in case the incoming packet is not recognized, i.e., $REC_{er}$ event occurs. On the basis of the reception mechanism previously described, we can express $w_0$ as

$$
w_0 = w_{RX}(AC) + w_{RX}(HEAD) \cdot \frac{P(HEC_{er} | REC_{er})}{P(REC_{er})}.
$$

where, as usual, $P(a)$ denotes the probability of the generic event $a$ and $P(a|b)$ the conditioned probability of $a$ given $b$.

Finally, when necessary, we add the superscript $(M)$ and $(S)$ to discriminate between master and slave unit. For example, $AC_{er}^{(S)}$ denotes the occurrence of an AC error event at the slave unit, during the reception of a downlink packet.

In order to identify the state space $E$ and the system dynamic associated to each state $E_i$, we look at the reception status of a pair of downlink (master to slave) and uplink (slave to master) packets. We can define the following four *basic events*:

$$
r_1 = \{REC_{ok}^{(S)} \cap REC_{ok}^{(M)} \};
$$

$$
r_2 = \{REC_{er}^{(S)} \};
$$

$$
r_3 = \{CRC_{er}^{(S)} \cap REC_{er}^{(M)} \};
$$

$$
r_4 = \{PR_{ok}^{(S)} \cap REC_{er}^{(M)} \};
$$

where the symbol $\cap$ denotes the *intersection function*. Event $r_1$, for example, occurs when both slave and master units successfully recognize the downlink and uplink packet, respectively. Event $r_3$, instead, occurs when the downlink packet is received by the slave, though with unrecoverable errors in the PAYL field, while the return packet is not recognized by the master.

It is easy to realize that the basic events are disjoint and their union covers all the possible reception cases for a pair of downlink and uplink packets. Furthermore, since the error statistics for successive packets are mutually independent, the probability of a basic event $r_i$ can be factorized in the product of the probabilities of the corresponding reception events at slave and master units. For example, we have $P(r_1) = P(REC_{ok}^{(S)}) \cdot P(REC_{ok}^{(M)})$.

On the basis of the reception mechanism described in Section II, each basic event determines the evolution of the master and slave units for a given number of successive slots, after which another event can occur independently of the past history of the system. Hence, we associate to each basic event $r_i$ a state $E_i$ of our FSMC. The state $E_i$ captures the system dynamic in the slots immediately following the occurrence of the basic event $r_i$. The system leaves the
Fig. 2. Example of system dynamic in state $E_1 (D_{xn} = DH3, D_{ym} = DH5)$.

Fig. 3. Example of system dynamic in state $E_2 (D_{xn} = DH5)$.

Fig. 4. Example of system dynamic in state $E_3 (D_{xn} = DH5, D_{ym} = DH3)$.

Fig. 5. Example of system dynamic in state $E_4 (D_{xn} = DH3, D_{ym} = DH3)$. 
state \( E_i \) when there is no further memory of the occurrence of \( r_i \) and another basic event \( r_j \) can occur with probability \( P(r_j) \).

Let \( D_{xn} \) and \( D_{ym} \), with \( n, m \in \{1, 3, 5\} \) and \( x, y \in \{H, M\} \), be the packet types used for downlink (from master to slave) and uplink (from slave to master) transmissions, respectively. Then, the system dynamic for each state \( E_i \in \mathcal{E} \) is as follows.

**State \( E_1 \)**

In state \( E_1 \), both downlink and uplink packets are recognized by the corresponding units. The master gets the acknowledgment (ACK) information related to the previous downlink transmission, from the HEAD of the uplink packet returned by the slave. In case of positive ACK, the master schedules the transmission of a new packet in the following downlink slot. Otherwise, the old downlink packet is scheduled for retransmission. At this point, a new basic event can occur and the state \( E_1 \) is left. An example of the system dynamic in state \( E_1 \) is shown in Fig. 2, where the shadowed area represents the active periods of master and slave units.

Assuming downlink and uplink packet are \( n \) and \( m \) slot long, respectively, the time the system spends in state \( E_1 \) is equal to
\[
T_1 = (n + m)T_{\text{slot}} \quad ;
\]
while the average amount of energy consumed by master and slave unit is given, respectively, by
\[
W^{(M)}_1 = w_{TX}(D_{xn}) + w_{RX}(D_{ym}) \quad ,
\]
\[
W^{(S)}_1 = w_{RX}(D_{xn}) + w_{TX}(D_{ym}) \quad .
\]

Finally, denoting by \( \mathbb{D}(x) \) the number of data bits carried by the PAYL of packet type \( x \), the average amount of data successfully delivered by master and slave unit is given by
\[
D^{(M)}_1 = \mathbb{D}(D_{xn}) \cdot P(PR_{ok}^{(S)} \mid REC_{ok}^{(S)}) \quad ,
\]
\[
D^{(S)}_1 = \mathbb{D}(D_{ym}) \cdot P(PR_{ok}^{(M)} \mid REC_{ok}^{(M)}) \quad ;
\]
where \( P(PR_{ok} \mid REC_{ok}) \) is the conditioned probability that PAYL field is successfully decoded given that the packet is recognized.

**State \( E_2 \)**

State \( E_2 \) is entered when the slave does not recognize the incoming packet and, hence, cannot reply to the master. The master, then, does not get any acknowledgment from the slave and schedules the packet for retransmission in the following (downlink) slot. This completes the system dynamic in state \( E_2 \). A graphical representation of this case, for five-slot long downlink packets, is shown in Fig. 3. We note that the number of times the slave wakes up listening for a valid AC, after the first recognition failure, is equal to \( \lfloor n/2 \rfloor \), where \( \lfloor \cdot \rfloor \) denotes the floor function. The master transmits the entire \( D_{xn} \) packet before switching in receive mode. Reception, however, stops immediately after the AC recognition has failed. Hence, we have that the time spent in state \( E_2 \) is equal to
\[
T_2 = (n + 1)T_{\text{slot}} \quad ;
\]
while the average amount of energy consumed by master and slave units is given
\[
W^{(M)}_2 = w_{TX}(D_{xn}) + w_{RX}(AC) \quad ,
\]
\[
W^{(S)}_2 = w^{(S)}_0 + w_{RX}(AC) \cdot \left\lfloor \frac{n}{2} \right\rfloor \quad .
\]
Since the downlink packet is not recognized, no data is successfully transferred in either directions, i.e.,

\[ D_2^{(M)} = 0, \quad D_2^{(S)} = 0. \]

**State E₃**

State \( E₃ \) is entered when the slave receives the incoming \( Dxₙ \) packet, though with unrecoverable errors in the PAYL field, while the master does not recognize the return \( Dym \) packet and schedules \( Dxₙ \) for retransmission (event \( r_3 \)). An example of the system dynamic in state \( E₃ \) is shown in Fig. 4, for \( Dxₙ = DH₅ \) and \( Dym = DH₃ \). As shown in the figure, in case \( Dym \) is multi-slot \((m > 1)\) the master will start retransmitting the \( Dxₙ \) packet before the slave has completed the transmission of the \( Dym \) packet. Clearly, the slave cannot receive any packet while it is transmitting and, hence, it will miss the master retransmission. It is easy to verify that, under the hypothesis of continuous transmission, the number of retransmissions attempted by the master before the end of the \( Dym \) transmission is exactly

\[ z = \left\lceil \frac{m - 1}{n + 1} \right\rceil, \tag{13} \]

where \( \lceil \cdot \rceil \) denotes the ceiling function. Note that, depending on the length of the packet types used, it is possible that the slave switches back in receive mode after the master has started the \( z \)-th retransmission of the downlink packet. In this case, the slave does not get a valid AC until the next master retransmission. Considering all the possible combinations of downlink and uplink packet types, we find that the number of times the slave unit wakes up scanning for a valid AC during the \( z \)-th retransmission is given by

\[ i^{(S)}(n, m) = \begin{cases} 2, & \text{for } n = 5, \ m = 3; \\ 1, & \text{for } n = m > 1; \\ 0, & \text{otherwise}. \end{cases} \tag{14} \]

When the \( z \)-th retransmission is completed the slave is ready to receive a packet from the master. The system dynamic in state \( E₃ \) can be considered, at this point, completed. The time duration of state \( E₃ \) is, then, given by:

\[ T_3 = (n + 1) (z + 1) T_{\text{slot}}. \tag{15} \]

while, the amount of energy consumed by master and slave units is given, respectively, by

\[ W_3^{(M)} = w_{TX}(Dxₙ) \cdot (z + 1) + w_0^{(M)} + w_{RX}(AC) \cdot z, \tag{16} \]

\[ W_3^{(S)} = w_{RX}(Dxₙ) + w_{TX}(Dym) + w_{RX}(AC) \cdot i^{(S)}(n, m). \tag{17} \]

Finally, since downlink packet is received with unrecoverable errors in the PAYL field, while uplink packet is not even recognized, the average number of useful data bit delivered in both directions is zero, i.e.,

\[ D_3^{(M)} = 0, \quad D_3^{(S)} = 0. \]

**State E₄**

State \( E₄ \) is entered when the downlink packet is perfectly received by the slave unit, while the return packet is not recognized by the master unit. The dynamic of the system in state \( E₄ \) follows, initially, the same steps as in state \( E₃ \): the master performs \( z \) packet retransmissions before the slave switches back in receive mode. At this point, however,
the situation differs from state \( E_2 \), in that the PAYL field of the downlink packet was successfully received by the slave. Consequently, successive master retransmissions leads to the reception of duplicate packets by the slave. On the basis of the retransmission mechanism described in Section II-C, whenever the slave recognizes a duplicate packet, it disregards the packet payload and returns an uplink packet which contains a positive acknowledgment for the downlink transmission. The master keeps retransmitting again and again the same packet until it recognizes an uplink packet, i.e., an \( r_1 \) event occurs. At this point, the system dynamic associated to state \( E_4 \) is concluded. Fig. 5 depicts an example of system dynamic in state \( E_4 \).

In order to evaluate the reward functions in state \( E_4 \), we identify four basic blocks, namely \( \alpha \), \( \beta \), \( \gamma \) and \( \delta \), which correspond to the different parts that the system dynamic in state \( E_4 \) is composed of.

Block \( \alpha \) describes the evolution of the system when state \( E_4 \) is entered. The system dynamic in this block is basically the same as in state \( E_3 \). Time duration and energy consumption in this block are, then, given by

\[
T_\alpha = (n + 1) (z + 1) T_{\text{slot}} ; \\
W^{(M)}_\alpha = w_{TX} (Dxn) \cdot (z + 1) + w_0^{(M)} + w_{RX} (AC) \cdot z , \\
W^{(S)}_\alpha = w_{RX} (Dxn) + w_{TX} (Dym) + w_{RX} (AC) \cdot i^{(S)}(n,m) .
\]

The amount of useful data bit delivered by the master and slave units is, instead, given by

\[
D^{(M)}_\alpha = D(Dxn) , \\
D^{(S)}_\alpha = 0 ;
\]

where (21) comes from the successful reception of the PAYL field by the slave unit, while (22) derives from the recognition failure of the return packet by the master.

Block \( \beta \) describes the dynamic of the system when a master retransmission incurs in an \( r_2 \) error event. In this case, the system follows exactly the same steps as in state \( E_2 \). Hence, the rewards earned in this phase are the same as in \( E_2 \), i.e.,

\[
T_\beta = (n + 1) T_{\text{slot}} ; \\
W^{(M)}_\beta = w_{TX} (Dxn) + w_{RX} (AC) , \\
W^{(S)}_\beta = w_0^{(S)} + w_{RX} (AC) \cdot \lceil \frac{n}{2} \rceil ; \\
D^{(M)}_\beta = 0, D^{(S)}_\beta = 0 .
\]

Block \( \gamma \) is associated to any further occurrence of either event \( r_3 \) or \( r_4 \), after the execution of block \( \alpha \). In this case, the downlink packet is recognized, while the return packet is not. The system follows the same steps as described in state \( E_3 \), except that the slave stops receiving immediately after the recognition of the HEAD field of the downlink packet, disregarding the PAYL field. The time duration of this block and the average amount of energy consumed by the master unit are, then, the same as in state \( E_3 \), i.e.,

\[
T_\gamma = (n + 1) (z + 1) T_{\text{slot}} ; \\
W^{(M)}_\gamma = w_{TX} (Dxn) \cdot (z + 1) + w_0^{(M)} + w_{RX} (AC) \cdot z .
\]
Since the slave stops receiving before the PAYL field, the average amount of energy consumed by the slave unit is

\[ W^{(S)}_{\gamma} = w_{RX}(AC) + w_{RX}(HEAD) + w_{TX}(Dym) + w_{RX}(AC) \cdot i^{(S)}(n, m). \]

Furthermore, the amount of useful data delivered by the two units is zero because the downlink packet carries duplicate data, while the uplink packet is not recognized, thus,

\[ D^{(M)}_{\gamma} = 0, \quad D^{(S)}_{\gamma} = 0. \]

Block \( \delta \) describes the evolution of the system just before state \( E_4 \) is left, i.e., when the event \( r_1 \) occurs. The system dynamic is similar to that of state \( E_1 \), except for the slave that disregards the PAYL field of duplicate packets. Therefore, the rewards earned in this phase are the following:

\[ T^{(M)}_{\delta} = (n + m) \cdot T_{slot}, \quad \text{(28)} \]
\[ W^{(S)}_{\delta} = w_{TX}(Dxn) + w_{RX}(Dym), \quad \text{(29)} \]
\[ W^{(M)}_{\delta} = 0, \quad \text{(30)} \]
\[ D^{(M)}_{\delta} = 0, \quad \text{(31)} \]
\[ D^{(S)}_{\delta} = \mathbb{D}(Dym) \cdot P \left( PR^{(M)}_{ok} \big| REC^{(M)}_{ok} \right). \quad \text{(32)} \]

Let \( \tau \) be the number of times the system iterates on block \( \beta \) and \( \gamma \) before the event \( r_1 \) occurs and the system enters in block \( \delta \). Hence, \( \tau \) is a geometrically distributed random variable, having probability distribution function given by

\[ P(\tau = k) = P(r_1) \cdot (1 - P(r_1))^k; \quad k = 0, 1, \ldots, \quad \text{(33)} \]

and expectation \( \bar{\tau} \) equal to

\[ \bar{\tau} = \frac{1 - P(r_1)}{P(r_1)}. \quad \text{(34)} \]

Let \( n_\beta \) and \( n_\gamma \) be the number of times the system cycles on blocks \( \beta \) and \( \gamma \), respectively, before entering the block \( \delta \). Hence, for a given value of \( \tau \), we have

\[ n_\beta = \sum_{j=0}^{\tau} \chi \{ j \text{-th iteration on block } \beta \} ; \quad \text{(35)} \]
\[ n_\gamma = \sum_{j=0}^{\tau} \chi \{ j \text{-th iteration on block } \gamma \} ; \quad \text{(36)} \]

where \( \chi \{ x \} \) is the indicator function of the event \( x \). Taking the expectation of both sides of (35), we get

\[ \bar{n}_\beta \doteq E \left[ n_\beta \right] = E \left[ \tau \right] P \left( r_2 \mid r_2 \cup r_3 \cup r_4 \right) \]
\[ = \frac{P(r_2)}{P(r_1)} = \frac{1 - P_{S}}{P_{S}P_{M}}; \quad \text{(37)} \]

where, for ease of reading, we have denoted the probabilities \( P(REC^{(S)}_{ok}) \) and \( P(REC^{(M)}_{ok}) \) by \( P_{S} \) and \( P_{M} \), respectively. Analogously, taking expectations of (36), we obtain

\[ \bar{n}_\gamma \doteq E \left[ n_\gamma \right] = E \left[ \tau \right] P \left( r_3 \cup r_4 \mid r_2 \cup r_3 \cup r_4 \right) \]
\[ = \frac{P(r_3) + P(r_4)}{P(r_1)} = \frac{1 - P_{M}}{P_{M}}, \quad \text{(38)} \]
We can, finally, compute the average reward earned by the system in state \( E_4 \). Denoting by \( A \) the generic reward function, and by \( A_h \) the amount of reward earned in block \( h \in \{ \alpha, \beta, \gamma, \delta \} \), we have

\[
A_4 = A_\alpha + \bar{\beta} A_\beta + \bar{\gamma} A_\gamma + A_\delta .
\] (39)

Solving (39) for \( A = T \), \( A = W \), and \( A = D \), we obtain, after some algebra, the following reward values:

\[
T_4 = \left( n + m + (n + 1) \frac{z P_s + 1}{P_M P_s} \right) \cdot T_{\text{slot}} ;
\] (40)

\[
W_{4}^{(M)} = w_{TX}(Dxn) \left( \frac{z P_s + 1}{P_M P_s} + 1 \right) + W_{RX}(AC) \left( \frac{P_s (z - 1) + 1}{P_M P_s} \right)
\] (41)

\[
+ \frac{w_0^{(M)}}{P_M} + w_{RX}(Dym) ,
\]

\[
W_{4}^{(S)} = w_{RX}(Dxn) + w_{RX}(AC) \left( \frac{\iota^{(S)}(n, m) + 1}{P_M} + \left\lfloor \frac{n}{2} \right\rfloor \frac{1 - P_s}{P_s P_M} \right)
\] (42)

\[
+ w_{RX}(\text{HEAD}) \frac{P_M}{P_M} + w_{TX}(Dym) \left( 1 + \frac{1}{P_M} \right) + w_0^{(S)} \frac{1 - P_s}{P_s P_M} ;
\]

\[
D_{4}^{(M)} = \mathbb{D}(Dxn) ;
\] (43)

\[
D_{4}^{(S)} = \mathbb{D}(Dym) \frac{P(PR_{ok}^{(M)})}{P_M} .
\] (44)

Concluding, the average amount \( \bar{A} \) of reward \( A \) earned for each transition of the FSMC is given by

\[
\bar{A} = \sum_{E_j \in E} \pi_j \cdot A_j = \sum_{j=1}^{4} P(r_j) \cdot A_j
\] (45)

\[
= P_s P_M A_1 + (1 - P_s) A_2 + P(CRC_{er}^{(S)})(1 - P_M) A_3 + P(PR_{ok}^{(S)})(1 - P_M) A_4 .
\]

Replacing \( A \) with \( T \), \( W \) and \( D \) in (45), we obtain the average amount of elapsed time \( \bar{T} \), energy consumed \( \bar{W} \), and useful data transfer \( \bar{D} \), for each transition of the FSMC.

**IV. PERFORMANCE ANALYSIS**

In this section we analyze the performance achieved by various Bluetooth packet formats, in different cases. Throughout the following analysis, we consider the bit error rate (BER) statistic provided in [15] for a specific, though very common, GFSK transceiver. We assume that the average Signal to Noise Ratio (SNR) value is the same for the master and slave units. Furthermore, we normalize to 1 the amount of energy required for transmitting or receiving a bit, i.e., we consider \( P_{TX} = P_{RX} = 1 \).

**A. Performance metrics**

System performance is evaluated in terms of the goodput (\( \mathcal{G} \)) and energy efficiency (\( \xi \)).

The goodput \( \mathcal{G} \) provides a measure of the average transmission capacity that the baseband layer offers to the higher protocols and is defined as the average amount of successfully delivered data bits per unit of time. Therefore, the overall system goodput can be obtained as:

\[
\mathcal{G} = \frac{\bar{D}^{(M)} + \bar{D}^{(S)}}{\bar{T}} .
\] (46)
The energy efficiency $\xi$ is defined as the average amount of successfully delivered data bit per unit of energy [16]. Thus, the overall system efficiency is defined by

$$\xi = \frac{\bar{D}^{(M)} + \bar{D}^{(S)}}{\bar{W}^{(M)} + \bar{W}^{(S)}}.$$  \hspace{1cm} (47)

B. Performance of different packet types

Protecting payload with FEC produces two opposite effects. On the one hand, the FEC theoretically improves $CRC_{er}$ and lowers $PR_{er}$. On the other hand, the code overhead reduces the payload capacity. Thus, a trade–off between goodput realized by protected ($DM_n$) and unprotected ($DH_n$) packet types may be expected.

AWGN channel

Let us denote by $(M \succ S)$ the configuration with only downlink data traffic, and by $(S \succ M)$ the reverse configuration, with only uplink data traffic. Fig. 6 and Fig. 7 show the average system goodput and energy efficiency, respectively, versus SNR, for a $(M \succ S)$ connection in an AWGN channel. The six curves have been obtained by changing the packet type used by the master unit, while the slave unit always used NULL packets. A first evidence that arises from the figures is that for SNR values lower than $\sim 14$ dB, the system goodput is practically zero and communication would probably be impossible. For SNR 18 dB, best performance, both in terms of goodput and energy efficiency, is achieved by unprotected packet types, namely $DH_5$ and $DH_3$. On the contrary, protected formats ($DM_5$, $DM_3$) appear more suitable for SNR values lower than 18 dB. Finally, we can see that single–slot packet types achieve very low performance for all the values SNR considered. Swapping the roles of master and slave units, i.e., considering an $(S \succ M)$ asymmetric uplink connection, system performance shows some variation. In Fig. 8, we plot the following goodput ratio:

$$\Delta G = \frac{G(S \succ M)}{G(M \succ S)};$$  \hspace{1cm} (48)

while Fig. 9 shows the energy efficiency ratio, given by:

$$\Delta \xi = \frac{\xi(S \succ M)}{\xi(M \succ S)}.$$  \hspace{1cm} (49)

We can observe that goodput achieved by $DM_3$ and $DM_5$ packet types in $(S \succ M)$ configuration is up to 15% higher than in $(M \succ S)$ configuration. The maximum goodput improvement is obtained for SNR around 15.5 dB, while
for lower values performance gets worse. Furthermore, for SNR $\tilde{\xi} \approx 14$ dB, protected packet types show up to 20% performance improvement in terms of energy efficiency. Unprotected packet formats, on the contrary, achieve lower goodput values, in particular for SNR < 16 dB. In this SNR region, however, unprotected types are not suitable, since they achieve a very low goodput also in the $(M > S)$ configuration, as shown by Fig. 6.

Fading channel

Most of the scenarios envisioned for Bluetooth consists of indoor–environments, like offices, conference rooms, cafeterias, cars, and so on. In this case, a fading channel model may result more appropriate to describe the characteristics of indoor radio propagation. Curves shown in Fig. 10 give the average system goodput vs SNR for a $(M > S)$ asymmetric connection in a Rayleigh–fading scenario. The energy efficiency curves, related to the same scenario, are plotted in Fig. 11. As expected, system experiments a drastic performance loss in a Rayleigh channel. We can note that the $DH5$ packet type achieves higher goodput and energy efficiency for almost all the SNR values, even though, for SNR < 16 dB, DM5 packet type achieves slightly better performance. In any case, for SNR < 16 dB, the performance gap between protected and unprotected packet types is drastically reduced. Performance comparison between $(M > S)$ and $(S > M)$ asymmetric connections gives some interesting results. Fig. 8 and Fig. 9 show $\Delta G$ and $\Delta \tilde{\xi}$ against SNR. We can observe that, in case of Rayleigh fading channel, $(S > M)$ configuration yields much higher performance, both
in terms of goodput and energy efficiency, than \((M > S)\) configuration.

C. The receiver–correlator margin \((S)\)

The receiver–correlator margin \(S\) is an important design parameter that may strongly impact on system performance. The lower the \(S\) value, the higher the SNR required to get a packet accepted. On the one hand, a low value of \(S\) may help to limit the power that gets wasted by receiving payload–corrupted packets. On the other hand, an high value of \(S\) may increase the probability of correct reception of the packet header, which contains the ACK information. This may prevent the transmission of duplicate packets, saving energy and capacity. The value of \(S\), however, cannot be excessively increased since the probability of an erroneous trigger increases as well. An erroneous trigger event may occur in two cases: i) the correlator output exceeds the threshold before the complete reception of the sync word (misalignment); ii) the correlator is triggered by a not–expected sync word (violation). Generally, \(S\) being equal, the probability of misalignment is lower than the probability of violation. This is due to the auto correlation properties of the Barker sequence and the PN sequence (used to construct the sync word in the AC field) that guarantee a large Hamming distance between a sync word and any shift of the word itself. Hence, given that sync words based on different LAPs have minimum Hamming distance of \(d_{\text{min}} = 14\) bits, an upper bound for \(S\) may be set to \(d_{\text{min}}/2 - 1 = 6\).

In order to evaluate the impact of the receiver correlator margin \(S\) on system performance, we have fixed the SNR value to 15 dB and we have computed the performance indexes for \(S = 0, 1, \ldots, 6\). Results have been normalized with respect to \(S = 0\). Fig 14 and Fig 15 show graphs obtained for the AWGN and Rayleigh fading channel model, respectively. Goodput curves are plotted on the left hand side of each figure, while energy efficiency curves are on the right hand side. Fig 14 reveals that, for the AWGN channel, the performance achieved by the system improves for \(S > 0\). As expected, performance improvement is more relevant for short and protected packet types. A slightly different behavior is shown by \(DH5\) and \(DH3\) types, which achieve the best energy efficiency for \(S = 1\). The performance gain is reduced in case of Rayleigh fading channel, as appears by the graphs of Fig 15. Also in this case, the goodput achieved by the six packet types increases with \(S\). However, \(DH5\) and \(DH3\) packet types achieve their maximum energy efficiency for \(S = 0\) and \(S = 1\), respectively. Nevertheless, it may result convenient to set \(S = 6\), since the energy efficiency loss is on the order of 5%, while the goodput gain is approximately on the order of 30%.

The impact of \(S\) on system performance, however, rapidly reduces for higher values of SNR.
V. CONCLUDING REMARKS

In this paper, a detailed performance analysis of a point-to-point Bluetooth link was presented. We provided a simple mathematical model for the Bluetooth system dynamic, based on a finite state Markov chain. Hence, we applied the theory of renewal reward processes to derive the performance of the system in terms of goodput and energy efficiency.

The mathematical model has allowed us to study in details the system behavior in different environmental conditions and to investigate the potential performance tradeoff among various ACL packet formats. Furthermore, the impact on system performance of an important design parameter, namely the receiver–correlator margin $S$, has been also considered.

The study has revealed the presence of a tradeoff between average traffic rate achieved by different packet types. As expected, unprotected and long types yield better throughput and goodput for $\text{SNR} > 18$, while in worse channel conditions better performance are achieved by short and protected formats. However, in presence of fading, the gap between performance achieved by protected and unprotected formats is drastically reduced, in particular for low SNR values.

Furthermore, in case of asymmetric data transfer, better performance is achieved by configuring as slave the unit that hosts the server and as master the unit that hosts the client application, respectively. This configuration yields performance improvement in terms of both goodput and energy efficiency, since the server never retransmits packets that were already received by the client.

Finally, the choice of $S$ has shown to be critical, since it may significantly impact on performance achieved by short
and protected packet types, although long and unprotected packet types show less dependence on this parameter.

Although our analysis was focused on a piconet with only two units, the mathematical model we propose can be easily extended to more complex piconet structures. The results obtained in terms of energy efficiency and average goodput may, then, be exploited to design energy–efficient algorithms for the piconet management.

REFERENCES