Carrier–Sense ARQ: Squeezing Out Bluetooth Performance while Preserving Standard Compliancy

Andrea Zanella
Department of Information Engineering, University of Padova
Via Gradenigo 6/B, Padova, 35131, Italy
e-mail: zanella@dei.unipd.it

Abstract—In this paper, we propose a simple and standard compliant retransmission mechanism, called Carrier–Sense Automatic Repeat reQuest (CS–ARQ), which aims at improving system performance, both in terms of throughput and energy efficiency, by avoiding useless data packet retransmissions. More specifically, in case of missed acknowledgment (ACK), the source makes use of its carrier-sensing capabilities to decide whether retransmitting the data packet or soliciting the ACK retransmission from the destination. The scheme is modeled by means of a two–state Markov Chain that permits to determine closed-form expressions for the throughput and energy efficiency figures. The analysis reveals that the CS–ARQ mechanism is actually capable of significantly enhancing the system performance, in particular in some critical scenarios, while preserving standard compliancy.

I. INTRODUCTION

With the introduction of the Enhanced Data Rate (EDR) mode, Bluetooth aims at reaching out to traffic-intensive applications for wireless personal area networks. However, enabling this type of services requires to squeeze as much performance as possible out of Bluetooth specifications, both in terms of throughput and energy efficiency [1]. As a matter of fact, the actual performance achieved by most wireless systems in realistic scenarios is often much worse than the nominal one, due to a number of limiting factors, such as protocol overheads, interference, noise, and so on. To limit the effect of these factors, Bluetooth specifications include several countermeasures at different layers of the protocol stack. In particular, the baseband layer provides different modulation schemes and frame formats, together with a mandatory Automatic Repeat-reQuest (ARQ) mechanism and an optional Forward Error Correction (FEC) scheme that aim at improving the reliability of the data transmission. In this paper we propose a simple and standard compliant retransmission mechanism, called Carrier–Sense Automatic Repeat reQuest (CS–ARQ), which aims at improving Bluetooth performance by avoiding useless packet retransmissions. To this end, CS-ARQ freezes the retransmission process and enters a soliciting state any time the sender senses a busy channel without receiving a valid frame. In the soliciting state, the sender keeps transmitting short control frames (POLL) to the destination, until it gets a valid (positive or negative) ACK in return. At this point, the sender resumes the original ARQ process as if the soliciting state was never entered. CS-ARQ performance is thoroughly investigated by means of a mathematical model, based on a two–state Markov Chain, that permits to determine closed-form expressions for the throughput and energy efficiency of a link.

II. RELATED WORK

The problem of performance maximization in Bluetooth system has been tackled by some recent works. In [3], different Segmentation and Reassembly (SAR) strategies and scheduling algorithms for BT are compared. Custom coding and adaptive frame format control are investigated in [4] whereas [5] proposes the use of turbo coding to enhance the link–layer throughput for Bluetooth v1.2. Hybrid ARQ/FEC schemes have been investigated in [6], [7]. The energy efficiency of some error control schemes in Bluetooth-based wireless sensor networks has been investigated in [8], where different FEC schemes and adaptive techniques are compared in multi-hop scenarios. Some discussions on the energy efficiency aspects of the Bluetooth system can also be found in [9] and references therein. Such works are mainly focused on the definition of dynamic power management policies, which reduce the energy consumption by using the low power modes provided by the standard.

Most of the techniques proposed in the cited papers aim at alleviating the performance loss due to channel impairments. The achievement of this goal is often pursued by dint of the introduction of significant modifications to the Bluetooth standard and/or more complex hardware. Conversely, CS-ARQ simply attempts to make a better use of the native tools provided by the system. Therefore, not only CS-ARQ is fully standard compliant, but it can also work in parallel with any other performance-enhancement mechanism that is compatible with the native ARQ algorithm.

We apply such a model to a proof of concept scenario, which reveals that the CS–ARQ mechanism is actually capable of significantly enhancing the system performance at the rather limited cost of enabling the devices to perform carrier–sensing, a features that is generally provided by last generation chipsets [2]. The remainder of this paper is organized as follows. Sec. II overviews the related work. Sec. III provides an summary of the main features of Bluetooth standard. Sec. IV describes the CS-ARQ scheme and the rationale behind it. The structure and transition probabilities of the Markov chain that models the protocol behavior are given in Sec. V, where we also derive the reward functions and the performance metrics of the system. Sec. VI presents the performance analysis of the mechanism in a case study. Finally, Sec. VII concludes the paper with some final remarks.
III. BLUETOOTH BASICS

This section shortly overviews the features of the Bluetooth technology that are of interest for our analysis. Readers interested on the details are referred to the official standard [10].

The basic Bluetooth network configuration is the so-called piconet, a cluster of no more than eight devices sharing a common frequency-hopping radio channel. When the piconet is established, one device gets the master role, while the others get the slave role. Direct communication can occur between slaves and master only, according to a star topology with the master in the center. Full-duplex communication is obtained through Time-Division Duplexing (TDD): time is divided in consecutive slots of \( T_s = 625 \mu s \) that are alternatively used for master’s and slaves’ transmissions. Bluetooth encompasses both voice and data links. However, in this paper we consider Asynchronous Connection-Less (ACL) links only, which are intended for non-realtime data traffic. ACL frames begin with the access code (AC), used for signal acquisition, followed by the packet header (HEAD) that carries control information, including the frame type, the destination address, and an acknowledgment flag (ARQN). AC and HEAD are always transmitted at the basic rate \( R_1 = 1 \) Mbps. The upper layer data is carried in the payload field (PAYL), which can be transmitted at either one of the three rates provided by Bluetooth v2.0+EDR, i.e., \( R_1 = 1 \) Mbps, \( R_2 = 2 \) Mbps or \( R_3 = 3 \) Mbps. Frames transmitted at Enhanced Data Rate (EDR) also include a guard time, a synchronization and a trailer fields. In any case, the time occupancy of ACL frames is limited to 1, 3 or 5 consecutive time slots. The different frame formats will be denoted by \( jDHi \), where \( j = 1, 2, 3 \) is the slot occupancy rate (in Mbps), while \( n = 1, 3, 5 \) is the slot occupancy. Occasionally, 2EDR and 3EDR will be used to denote all the three frame formats transmitted at 2 Mbit/s and 3 Mbit/s, respectively. Besides data frames, the standard also encompasses two control frames, named POLL and NULL, that consist of AC and HEAD only, with no payload.

Channel access is managed by the master through a simple polling scheme: each master’s transmission commands the recipient slave to return a packet in the following slave’s slot. A mandatory Stop&Wait ARQ is used to improve communication reliability: data frames are retransmitted until the sender gets a positive acknowledgment (ACK) from the destination. The ACK information is piggybacked by the ARQN flag in the header of the return packet. Since negative ACK is implicitly assumed, the loss of an ACK will trigger retransmission even when the data frame was actually successfully delivered to the destination. In this case, the packet replications are coined duplicate packets (DUPCKs). Note that DUPCKs can be transmitted only by the master. In fact, slaves are enabled to transmit only upon reception of a valid master’s frame that, in turn, carries the (positive or negative) ACK for the previous slave-to-master transmission. Therefore, slave’s retransmissions occur only when receiving a master’s frame carrying a negative ACK.

IV. CARRIER-SENSE ARQ

The transmission of DUPCKs is detrimental to system performance, both in terms of throughput and energy efficiency, as noted in [4], [11], [12].

The CS-ARQ algorithm aims at alleviating this source of performance erosion by eliminating any chance for the master to transmit DUPCKs. To this end, CS-ARQ adopts a conservative approach, which can be summarized as follows: “Retransmit only in evidence of failure, otherwise ask the slave for feedback.”

The evidence of failure consists in the occurrence of either one of the following two events: i) the master receives a slave’s packet carrying a negative ACK; ii) no energy is sensed on the radio channel during the slave’s slot. The first condition is self-explaining, whereas the second condition arises from the fact that an idle slave’s slot is a clear indication that the slave did not return any packet because it was not able to recognize the AC or HEAD of the master’s frame. In either cases, the master’s frame was not successfully received and needs to be retransmitted, as dictated by the basic ARQ scheme. Conversely, if the master gets a slave’s frame carrying a positive ACK, it can proceed transmitting a new packet, still in accordance with the basic ARQ scheme. The only situation of uncertainty is when a frame is returned by the slave (and, hence, the channel is sensed busy) but the master cannot get the ACK flag because of unrecoverable errors in the reception of the AC or HEAD fields. In this case, the CS-ARQ scheme freezes the retransmission mechanism and starts a Soliciting phase to resolve the ambiguity. During this phase, the master keeps sending POLL frames to the slave, until it gets a valid ACK in return. At this point, the ARQ mechanism can be resumed and carried on according to the received ACK value.

As mentioned, CS-ARQ adopts a conservative approach, so that the soliciting phase is invoked even in case of missed detection of frames carrying negative ACK, which would ask for data retransmission. In this case, the soliciting phase yields extra delay and energy waste. Therefore, it is interesting to identify in which conditions benefits overcome drawbacks.

V. MATHEMATICAL MODEL

In this section, we first introduce some notation, then we describe the mathematical model of the CS-ARQ mechanism from which we derive the performance metrics of interest.

As explained in Sec. III, the reception of a baseband frame is performed in three consecutive steps corresponding to: 1) Access code acquisition (\( A \)); 2) frame Header recognition (\( H \)); 3) Data (payload) reception (\( D \)). If one of such step fails, then the unit enters power saving mode, skipping the remaining reception steps.

The success or failure of a given reception step will be denoted by \( P_X \). Hence, with reference to a single packet reception, the following relations hold

\[ P_{A_s} = 1 - P_{A_f} = P_{H_s} + P_{H_f}; \quad P_{H_s} = P_{D_s} + P_{D_f} \quad (1) \]

When needed, we add the superscript (\( M \)) or (\( S \)) to indicate that the event refers to a packet sent by the master or slave, respectively.
Note that, by virtue of the frequency hopping mechanism, successive frames are transmitted over different radio channels and, consequently, they are subjected to independent fading [13]. Therefore, the reception probabilities of master and slave are independent. For space constraints, we do not report here the expressions for all these probability functions, which are out of the scope of this paper. Instead, we refer the interested reader to the literature [14].

### A. Markovian Model

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 requires a more cumbersome notation and exposition, without adding any relevant concept.) We consider a heavy traffic scenario, where master and slave have always packets to transmit. We assume infinite retransmission limit: packets are retransmitted over and over again until the sender receives a positive acknowledgment. Furthermore, we assume that a single frame type is used in each direction, except that the master will transmit POLL frames in the Soliciting phase.

Under these assumptions, the dynamic of the system can be captured through a Markov Chain (MC) with state space $E = \{A, S\}$, where state $A$ is associated to the normal ARQ phase, whereas state $S$ is connected to the soliciting phase. The MC evolves in discrete steps, each corresponding to a single master-slave transmission phase. State transitions are governed by the CS-ARQ scheme. Let $\omega_n \in E$ be the state of the MC at the $n$-th step. Starting from $\omega_n = A$, the next step is $\omega_{n+1} = S$ when the master senses a busy channel but it is not able to decode the AC or HEAD of the slave’s frame. According to the notation previously introduced, the one-step transition probability $P_{AS}$ from $A$ to $S$ is then given by

$$P_{AS} = \Pr[\omega_{n+1} = S|\omega_n = A] = P^H_{H_s}(1 - P^{i,s}_{H_s})$$ (2)

In state $S$, the master keeps transmitting POLLs until it gets a valid ACK from the slave. When this happens, the MC steps back to state $A$ and the master resumes the ARQ mechanism. Hence, the one-step transition probability $P_{SA}$ from $S$ to $A$ is given by the probability $P^H_{H_s}$ that the master’s POLL is correctly received by the slave, times the probability $P^s_{H_s}$ that the master successfully decodes (at least) AC and HEAD of the returned frame

$$P_{SA} = \Pr[\omega_{n+1} = A|\omega_n = S] = P^H_{H_s} P^s_{H_s}$$ (3)

The steady state probabilities $\pi_A$ and $\pi_S$ of the MC being in states $A$ and $S$, respectively, are then given by

$$\pi_A = \frac{P_{SA}}{P_{AS} + P_{SA}} \quad \pi_S = \frac{P_{AS}}{P_{AS} + P_{SA}}$$ (4)

### B. Reward functions

Following the approach suggested in [15], the performance of the system can be investigated by resorting to a fundamental result of renewal reward theory, which states that the ratio between the rewards gained by the system in time asymptotically equals the ratio of the average rewards gained at every step of the MC.

We consider the following reward functions

- time reward ($T$): average time duration of a MC step;
- data reward ($D$): average data successfully delivered by a unit in a MC step;
- energy reward ($E$): average amount of energy consumed by a unit in a MC step.

In the following, $F_{j,n}$ and $F_{i,m}$ denote the type of packets used by master and slave, respectively, where $j$ and $i$ are the transmission rate (Mbit/s), whereas $n$ and $m$ denote the packet length (slot). POLL frames are denoted by $F_{poll}$.

#### Time Reward

In state $A$ the master transmits $F_{j,n}$ frames that take $n$ time slots each, whereas in state $S$ the master only transmits POLL frames that occupy a single slot each. If the slave recognises the AC and HEAD of the master’s packet, it replies with an $F_{i,m}$ frame that occupies $m$ time slots. On the contrary case, the slave is not allowed to transmit and the polling step ends after a single (empty) slot. Therefore, the average time reward “earned” by the MC per step is equal to

$$T = n \pi_A + \pi_S + P^{(i,s)}_{H_s} m + 1 - P^{(i,s)}_{H_s}$$ (5)

#### Data Reward

Let $D_{F_{\nu},k}$ denote the number of upper layer data bits carried by the PAYL field of a generic $F_{\nu,k}$ frame. In state $A$, the master transmits frames that have never been correctly received by the slave. Therefore, the successful reception of a master’s frame by the slave brings a data reward $D_{F_{j,n}}$. In the soliciting phase $S$, the master transmits POLL packets, which do not carry useful information and, hence, do not yield any data reward.

The slave unit, in turn, gains a data reward of $D_{F_{i,m}}$, when it first gets the possibility to transmit a frame by correctly decoding the header of the master’s packet and, second, its frame is successfully decoded by the master. Summing up, the average number of data bits successfully delivered by the master and slave units, respectively, in a MC step is given by

$$D = \pi_A D_{F_{j,n}} + \pi_S D_{F_{i,m}}$$ (6)

#### Energy Reward

The computation of the mean energy spent by the master and slave units per MC transition is simple, though cumbersome. Let $E^x_{X,Y}$ and $E^x_{X} \| E^y_{X}$ be the amount of energy consumed by a unit for transmitting, receiving and sensing, respectively, the generic packet (s) $X$. (For short, $X$ will take the value $A$, $H$, and $P$ for AC, HEAD and PAYL, respectively.) We start by considering the mean energy drained by the master unit. The energy spent in transmission is equal to $E^x_{F_{j,n}}$ when the system operates in state $A$ and to $E^x_{F_{i,m}}$ when it is in state $S$. The energy spent in reception depends on the events occurring during the data exchange. If the slave does not recognize the AC or the HEAD of the master’s frame, then the slave’s slot remains idle and the master spends only the energy required to sense the channel for a time period equal to the AC duration. Conversely, when an $F_{i,m}$ frame is returned, the master starts decoding the incoming signal, but immediately stops in case of missed detection of a valid AC or HEAD field. In this case, the master stops decoding and keeps sensing the channel till the
end of the slave’s transmission. Therefore, the average amount of energy spent by the master in a MC step is given by

$$\bar{E}^{(M)} = \pi_{A}E_{F_{\text{tx}}}^{(x)} + \pi_{S}E_{F_{\text{poll}}}^{(x)} + (P_{A_{j}} + P_{H_{j}})E_{A_{j}}^{(x)} + P_{H_{j}}E_{F_{\text{tx}}^{(x)}}^{(z)} + P_{H_{j}}^{(S)}E_{F_{\text{tx}}^{(z)}}^{(z)}$$

(7)

$$+ P_{H_{j}}^{(S)}E_{A_{j}}^{(z)} + P_{H_{j}}^{(S)}E_{H_{j}}^{(z)} + P_{H_{j}}^{(S)}E_{F_{\text{tx}}^{(z)}}^{(z)} + P_{H_{j}}^{(S)}E_{F_{\text{tx}}^{(z)}}^{(z)}$$

The energy spent by the slave unit can be obtained in a similar way, taking into consideration that, in state S, the master transmits POLL frames with no PAYL. Therefore, we get

$$\bar{E}^{(S)} = P_{H_{j}}^{(S)}E_{F_{\text{tx}}^{(x)}}^{(x)} + (P_{A_{j}} + P_{H_{j}})E_{A_{j}}^{(x)} + \pi_{S}E_{F_{\text{tx}}^{(x)}}^{(x)} + \pi_{S}E_{F_{\text{tx}}^{(x)}}^{(x)}$$

(8)

$$+ \pi_{S}E_{F_{\text{tx}}^{(x)}}^{(x)} + \pi_{S}E_{F_{\text{tx}}^{(x)}}^{(x)} + \pi_{S}E_{F_{\text{tx}}^{(x)}}^{(x)} + \pi_{S}E_{F_{\text{tx}}^{(x)}}^{(x)}$$

C. Performance metrics

Using the time, data and energy rewards we can obtain a number of different performance indexes. However, for space constraints we focus our attention on the Goodput \(G\) and Energy efficiency \(\xi\) metrics, defined as the average amount of successful delivered data bits per unit of time and per unit of energy [16], respectively, and given by

$$G = \frac{\bar{D}^{(M)} + \bar{D}^{(S)}}{T} \quad \xi = \frac{\bar{D}^{(M)} + \bar{D}^{(S)}}{\bar{E}^{(M)} + \bar{E}^{(S)}}$$

(9)

VI. CASE STUDY

For space limit, we present the results obtained in a single scenario consisting in a Rayleigh channel, with equal Signal-to-Noise ratio (SNR) at master and slave receiver and asymmetric (master-to-slave) data traffic. Energy figures have been obtained by assuming that the energy consumed in transmission, reception and channel sensing over a time interval \(T_s = 1\mu s\) is equal to \(P_{TX} \times T_s = 1\mu s\), \(P_{RX} \times T_s = 0.8\mu s\) and \(P_{SS} \times T_s = 0.1\mu s\), respectively, where \(\mu s\) (energy unit) is the amount of energy required for transmitting a bit at the basic rate [17]. Error probability functions have been taken from [14]. Note that the very same scenario was also considered in [11] for the performance analysis of the standard ARQ scheme, thus permitting direct comparison with CS-ARQ. However, to reduce clutter, we plot only the results for EDR data frame formats, which generally offer better performance than basic rate formats.

A. Performance analysis

Fig. 1 and Fig. 2 show the average system goodput and energy efficiency, respectively, obtained with the different EDR frame formats when the SNR varies. Dashed lines refer to the CS-ARQ goodput performance (\(G\)), whereas solid lines relate to the standard ARQ goodput performance (\(G^*\)) as derived in [11]. To better appreciate the comparison, in Fig. 3 we also plot the goodput gain \(\Delta G = G - G^*\), whereas Fig. 4 reports the relative energy efficiency gain \(\Delta \xi = (\xi - \xi^*)/\xi^*\) where \(\xi^*\) is the energy efficiency of the system with standard ARQ, as derived and plotted in [11]. Looking at the graphs, we can note that for SNR > 22 dB, best performance (both in terms of goodput and energy efficiency) is achieved by using 3DH5 frames, whereas 2DH5 packet format is preferable in the low SNR region. From the gain graphs, we can observe that the
most significant increase of CS-ARQ over standard ARQ is obtained with 2DH5 and 2DH3 frame formats, whereas the goodput gap for 3DH3 and 3DH5 formats is not exceptional, in relative terms, though still worth some tens of kbit/s for SNR > 20 dB. Conversely, CS-ARQ does not bring any goodput gain for single-slot formats, as it was expected since POLL frames have the same time occupancy of single-slot data frames without carrying any useful data. We can observe that CS-ARQ scheme yields from 5% up to 40% of energy gain with 2DH5 frames in the intermediate SNR region, where this type of frame achieves best performance. Some energy gain is also observed for single-slot frame formats. EDR formats, finally, experience a limited energy efficiency gain in the medium-to-high SNR range, whereas the energy efficiency worsens in the low SNR region where, as already noted, these frames achieve totally unsatisfactory performance also with standard setting. We observed similar results with other channel models, though the performance gap between CS-ARQ and standard ARQ becomes progressively less significant as the channel approaches AWGN.

VII. CONCLUSIONS

In this paper, we propose and analyze the CS-ARQ scheme, a simple and standard compliant retransmission mechanism for Bluetooth data links that aims at reducing the performance loss due to the transmission of DUPCKs by the master. As proof of concept, we compared the performance achieved by CS-ARQ against those obtained by the native ARQ mechanism in an asymmetric data connection, using EDR frame formats in Rayleigh fading channel. The study has revealed that CS-ARQ can yield appreciable goodput and energy efficiency gain with 2EDR frame formats, in particular in the intermediate SNR region where such packets perform better than any other. Conversely, the gain obtained with 3EDR frame formats is rather marginal, though still positive in the SNR region where such packets are more suitable. Instead, in the (unlikely) case that 3EDR formats were used in the low SNR region, then CS-ARQ would actually determine a small performance loss. Therefore, CS-ARQ shall be used in conjunction with a suitable rate adaptation algorithm that selects the most appropriate packet formats depending on the current channel conditions. In conclusion, CS-ARQ can provide a low-to-average performance gain that, though limited, is obtained by simply making a better use of the standard mechanisms defined by Bluetooth specifications. Furthermore, CS-ARQ can work in parallel with any other performance-enhancement mechanism compatible with the native ARQ algorithm, so that it can be considered as a general performance booster.

REFERENCES