Custom Error Control Schemes for Energy Efficient Bluetooth Sensor Networks

João H. Kleinschmidt, Walter C. Borelli and Marcelo E. Pellenz

Abstract — **This paper analyzes the effect of custom error control schemes on the energy efficiency in Bluetooth sensor networks. The energy efficiency metric considers in just one parameter the energy and reliability constraints of the wireless sensor networks. New packet types are introduced using some error control strategies in the AUX1 packet, such as Hamming and BCH codes, with and without CRC for error detection. Two adaptive techniques are proposed that change the error control strategy based on the number of hops traversed by a packet through the network. The performance results are obtained through simulations in a channel with Rayleigh fading for networks with different number of hops, showing that error control can improve the energy efficiency of a Bluetooth-based sensor network.**

Index Terms— **Bluetooth, error control, sensor networks, energy efficiency.**

I. Introduction

The recent advances in wireless communications and digital electronics led to the implementation of low power and low cost wireless sensors. A sensor node must have components for sensing, data processing and communication. These devices can be grouped to form a sensor network [1]. The medium access control (MAC) protocol is responsible for the creation of the network infrastructure and to share communication resources among the nodes. The suitability of existing MAC protocols for sensor networks have been studied and new ones are being developed specially for this purpose.

Bluetooth [2] is a low cost wireless radio technology designed to eliminate wires and cables between mobile and fixed devices over short distances. It operates on the 2.4 GHz ISM (Industrial, Scientific and Medical) band employing a frequency-hopping spread spectrum (FHSS) technique. The transmission rate is up to 1 Mbps, using GFSK (Gaussian Frequency Shift Keying) modulation. The channel is divided in time slots of 625 μs, using a time-division duplex (TDD) scheme for full-duplex operation.

The Bluetooth MAC protocol was designed to facilitate the formation of ad hoc networks. This characteristic makes the Bluetooth technology attractive for sensor networks, together with its low cost, multi-hop capabilities, device discovery process and energy saving modes. The devices can communicate with each other forming a network, called piconet, with up to eight nodes [2]. Devices in different piconets can communicate using a structure called scatternet. In [3] and [4] sensor networks were implemented using Bluetooth as the MAC protocol. Some protocols for scatternet formation and routing in Bluetooth sensor networks were proposed in [5], [6], [7] and [8].

The wireless channels have high bit error rates due to interference and the multipath propagation that characterizes the radio channel. In order to improve the reliability of the data sent in the wireless channel, many techniques can be employed, such as automatic repeat request (ARQ), forward error correction (FEC) or transmission power control. Although an error control strategy improves the reliability of a packet, the energy consumed due to the transmission of the additional bits in these coded schemes contributes to increase the energy consumption.

Some authors have studied the issue of energy consumption for some error control schemes in wireless sensor networks [9], [10], [11], [12]. In [9] and [10] the energy efficiency of different error control techniques was evaluated for sensor networks with a commercial radio transceiver using an analytical model. In [9] the energy efficiency is used as the metric for packet size optimization. In [10] the energy efficiency of some balanced channel codes is analyzed for different bit error probabilities. While in [11] the reliability and energy consumption were analyzed using simulation for sensor networks without any specific technology or channel model, in [12] the energy consumption and reliability of Bluetooth error control strategies were studied in a Rayleigh fading channel.

In this paper some novel error control schemes are proposed for Bluetooth sensor networks and a more general metric is used for performance analysis, the energy efficiency. This metric considers jointly the energy and reliability constraints of sensor networks. The effect of retransmissions and error control parities on energy efficiency is examined. New Bluetooth packet types are proposed using custom coding in the AUX1 packet. These modifications include a CRC code for error detection (without ARQ), BCH code with and without CRC and Hamming code with and without CRC. Two novel adaptive error control schemes that change the error control strategy accordingly to the number of hops traversed by a packet through the sensor network are introduced. The performance results were obtained through simulations in a channel with Rayleigh fading and for various sensor networks scenarios with different number of hops.

The techniques for error control of data packets, custom coding and adaptive schemes are presented in Section II. In Section III the simulation model based on Matlab is presented and Section IV presents the performance results

J. H. Kleinschmidt and W. C. Borelli are with the Department of Telematics, School of Electrical and Computer Engineering, State University of Campinas, UNICAMP, Campinas-SP, Brazil; (e-mail: {joaohk, borelli}@dt.fee.unicamp.br).

M. E. Pellenz is with the Graduate Program in Computer Science, Pontifical Catholic University of Paraná, PUCPR, Curitiba-PR, Brazil (e-mail: marcelo@ppgia.pucpr.br).

obtained. Finally, Section V gives the final considerations on which should be the best coding scheme for the Bluetooth-based sensor networks.

II. Error Control Strategies for Bluetooth Networks

A. Error Control of the Bluetooth Specification

The Bluetooth specification [2] defines seven asynchronous data packets. Each packet has three fields: the access code (72 bits), header (54 bits) and payload (0-2745 bits). The access code is used for synchronization and the header has information such as packet type, flow control and acknowledgement. The access code is error robust, because the coded synchronization words have a large Hamming distance (d_{min} = 14). The header contains a (n,k)=(3,1) repetition code for error verification. The payload carries the data bytes that usually are protected by an ARQ stop-and-wait strategy based in a CRC code. The receiver indicates in the next return packet whether the transmission was successful or not. The DMx packets have the data protected by a Hamming code (15,10) with rate 2/3. This code corrects all single bit errors and detects all two bits errors in a code word. Table 1 shows this information for each asynchronous packet.

TABLE I. ASYNCHRONOUS PACKET TYPES.

Packet	Time-slots	Payload (bytes)	FEC	CRC and ARQ
DM1	1	0-17	Hamming (15,10)	Yes
DH1	1	0-27	No	Yes
DM3	3	0-121	Hamming (15,10)	Yes
DH3	3	0-183	No	Yes
DM5	5	0-224	Hamming (15,10)	Yes
DH5	5	0-339	No	Yes
AUX1	1	0-29	No	No

A received packet is not accepted whenever one of the following events may happen: (A) the destiny fails to synchronize with the access code of the received packet; (B) the header of the received packet is corrupted (after the repetition code is decoded); (C) the data of the received packet are corrupted after the Hamming code is decoded, causing the CRC check to fail; (D) the source is unable to synchronize with the access code of the return packet and (E) the header of the return packet is corrupted. For packets without ARQ, only events A, B and C can cause errors, because such packets do not need the return packet to confirm the reception. The packet error probability of the forward channel, PER_f, and reverse, PER_r, can be defined [9] as:

$$PER_f = 1 - \int_0^\infty f(\gamma_f) P[\overline{A}] P[\overline{B}] P[\overline{C}] \, d\gamma_f \qquad (1)$$

$$PER_r = 1 - \int_0^\infty f(\gamma_r) P[\overline{D}] P[\overline{E}] \, d\gamma_r \qquad (2)$$

where $f(\gamma_f)$ and $f(\gamma_r)$ are the probability density functions and γ_f and γ_r are the signal-to-noise ratio (SNR) of the forward and reverse channels, respectively.

B. Custom Error Control

Whereas the packets defined by the Bluetooth standard (Table I) have fixed error control schemes, a custom coding can be implemented by making use of the AUX1 packet [12] [13]. With the AUX1 packet the Bluetooth device delivers the received bits independently whether they are correct or not. While the former asynchronous packets with ARQ maintain a reliable link with random delay (which approaches infinity for low values of SNR), the AUX1 packet may alternatively provide an unreliable link with delay of only one time slot.

It has been proposed in [13] the use of BCH codes with the CRC code for error detection. As the ARQ is turned off, it must be implemented at the application layer. The coder is implemented inserting a (232, k) BCH code in the payload of the AUX1 packet. The inputs of the BCH coder are the data and two CRC bytes, resulting in a packet with $K=k-16$ data bits. In order to accept the packet, the events A, B and C must not occur. The code then considered was a (232, 156) binary BCH code with a correction capability of up to t=10 errors.

It is being proposed in here some novel modifications in the AUX1 packet. The same BCH code can be applied, but without retransmission (BCH2 and BCH3 packets). Although this strategy can decrease the reliability of transmitted packets, in terms of energy consumption it can be very useful, for it is not necessary to send a return packet to indicate the success of the transmission. The BCH2 packet utilizes the CRC code for error detection, without asking retransmission. A packet is discarded if the CRC detects any errors. The BCH3 packet does not use either retransmission or CRC. The difference between BCH2 and BCH3 is that in the latter the packets are transmitted to the next node (in a multihop network) even if it contains errors, so wasting energy. In the BCH2 packet this fact does not happen, but the packet has additional 16 bits for the CRC implementation.

Another modification proposed in this work is to use the same Hamming code of the DMx packets in the AUX1 payload, but without retransmission, with and without CRC (HAM and HAM2 packets, respectively). Other new packet is the AUX2, which is an AUX1 packet with CRC code. Table 2 shows the error control information for the new introduced packet types.

TABLE II. PACKET TYPES WITH CUSTOM ERROR CONTROL.

Packet	Time-slots	Data (bytes)	FEC	ARQ	CRC
AUX2	1	0-27	No	No	Yes
HAM	1	0-18	Hamming (15,10)	No	No
HAM2	1	0-18	Hamming (15,10)	No	Yes
BCH	1	0-17	BCH (232,156)	Yes	Yes
BCH2	1	0-17	BCH (232,156)	No	Yes
BCH3	1	0-17	BCH (232,156)	No	No

C. Adaptive Error Control

Using the same error control scheme for the whole network could be a good choice in some cases, but not always. Sometimes it is needed to apply the best error control available, while in other cases less error control should be used. To use an adaptive error control scheme, a mechanism have to be designed to judge the importance of a packet and then choosing the most efficient error control scheme for that particular packet. In Bluetooth case, to change the error correction scheme means to change the

packet type to be transmitted. In order to apply an adaptive scheme in a sensor network, where the most important issue is to reduce the energy consumption, it was used the following approach similar to the proposed in [11] and [12].

The importance of a packet is evaluated using the multihop principle, as shown in Fig. 1. The choice of the packet type and the respective error control technique shall be based on the number of hops the packet traveled within the sensor network. For instance, a sensor node sends a data packet containing the information of the temperature of a small region to the sink node, which collects the data of all the sensors of the network.

However, before the packet reaches the sink node, it may travel through some other nodes of the network that can be sensors or another type of node with routing capacity. For instance, if the packet gets lost at the first hop, only the energy to send the packet from a sensor to a specific node was lost. If the packet instead is corrupted after few more hops, much more energy would be spent to transmit the packet through the network. In this sense, a packet is more important if it travels through more nodes in the network, and consequently, more energy is being consumed. An adaptive scheme might use stronger error control techniques for packets that travel more hops and weaker error control for packets with fewer hops.

In the adaptive error control scheme, each packet must have a counter with the number of hops the packet had in the network. This can be implemented as a field in the payload of the packet. Two different adaptive schemes were used: ADP1 and ADP2. A packet with weaker error control is used for the initial hops and a packet with more powerful coding for the remaining hops throughout the sensor network. Table III shows the packet types proposed in these schemes. Although only two schemes are being presented here, other adaptive strategies with different packet types might well being proposed.

TABLE III. ADAPTIVE SCHEMES.

Scheme	1st and 2nd Hops	3rd, 4th and 5th Hops	Other Hops
ADP1	AUX2	HAM2	DH1
ADP2	AUX2	BCH2	DH1

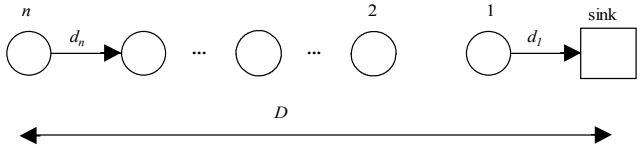

Fig. 1. Multi-hop sensor network.

III. SIMULATION MODEL

The simulations were implemented using the Matlab® software and are described in the sequence. The network considered is shown in Fig. 1, where a sensor must send data to the sink node. This is only one of many possible structures within the sensor network, which can have different topologies. It is being assumed that the Bluetooth scatternet was formed up and that the scheduling policy and the routes are also defined, using protocols as the proposed in [5], [6], [7] and [8]. The packet data is generated by a sensor node, that sends it to the next node, and so on, until it reaches the sink node. The wireless channel is modeled using the Rayleigh fading, whose probability density function is given by:

$$f(\gamma) = \frac{1}{\bar{\gamma}} \exp\left(-\frac{\gamma}{\bar{\gamma}}\right), \quad \text{for } \gamma \geq 0 \tag{3}$$

where $\bar{\gamma}$ is the average received signal-to-noise ratio (SNR) and γ is the instantaneous SNR.

Using equation (3) in (1) and (2) the error probabilities for each packet may be evaluated. These probabilities are given as a function of the signal-to-noise ratio. When a node receives a packet it is verified whether errors have occurred in the reception. If there were no errors the packet is sent to the next node. In the packets with ARQ, an acknowledgement is sent to the transmitter indicating the success of the transmission. On the other hand, if errors are detected, three actions can occur, depending of the packet type. In the packets with ARQ it is sent to the transmitter a packet indicating unsuccessful transmission (negative acknowledgement), so the packet will be sent again. In the packets without ARQ, the packet is discarded (with CRC) or sent to the next node (no CRC). It is important to note that the NULL packet used to acknowledge or not a transmission can also be corrupted, although it do not carry any data except the access code and header field. If the NULL packet is corrupted the node has to send the data packet again. Figures 2 and 3 show the flowcharts of the simulation model for the transmission and reception of packets in a sensor network environment.

For instance, consider the transmission of a DH1 packet, which has CRC and ARQ (Fig. 2). The sensor node sends the packet to the next node and then receives the return packet. If the return packet indicates negative acknowledgement the sensor has to send the DH1 packet again. If the return packet indicates successful transmission (acknowledgement) the node that received the DH1 packet will send it to the next node, until the DH1 packet reaches the sink. This process is repeated for each packet the sensor node has to send.

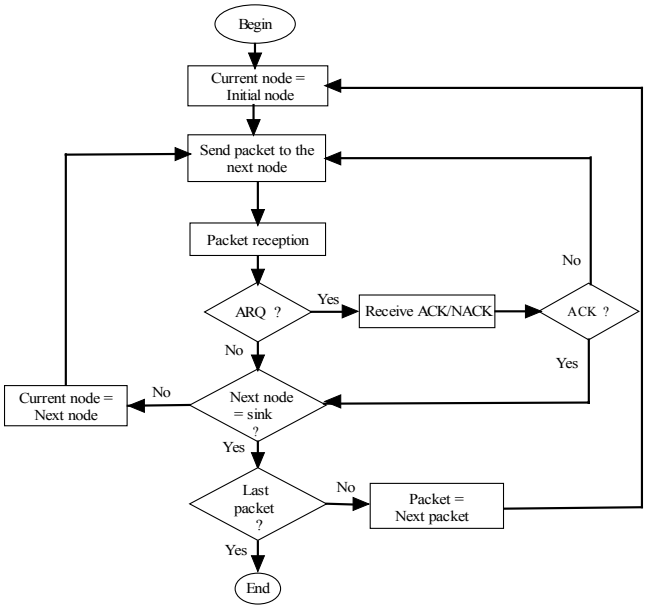

Fig. 2. Packets transmission.

At the receiver (Fig. 3), the CRC code verifies whether the packet contains errors or not. If errors were detected, the receiver asks the retransmission of the DH1 packet, sending

a negative acknowledgement. If no errors were detected, the receiver sends the acknowledgement of the DH1 packet.

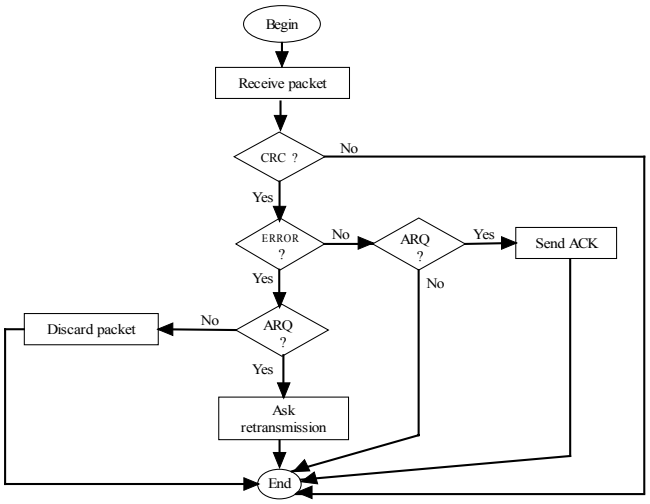

Fig. 3. Packets reception.

Since any specific hardware is being used, the energy consumption is expressed only in normalized terms. The energy considered are the energies spent in the transmission and reception of the packets. The energy to encode or decode a packet was not considered [9], [11], [14]. The coding is generally simpler and has less consumption than decoding. It is also considered that the Bluetooth device is in the connected state. More energy savings can be made during idle times using the power management schemes of Bluetooth (hold, park and sniff). Considering the same model as in [11] and [12], where the reception of a determined number of bits consumes approximately 75 per cent of the energy spent to transmit the same number of bits, the total consumed energy E is given by:

$$E = n_{bits} \times T_p + n_{bits} \times R_p \times 0.75 + n_{ack} \times P_a + n_{ack} \times R_a \times 0.75 \quad (4)$$

where n_{bits} is the total number of bits of a packet (access code, header and payload); T_p is the total number of transmitted packets (including retransmissions); R_p is the total number of received packets (including retransmissions); n_{ack} is the total number of bits of the return packet (NULL) and T_a and R_a the number of the return packets transmitted and received, respectively.

The reliability is given by the percentage of the sent packets being delivered correct to the sink node. Let n_{pac} be the total number of packets transmitted by the sensor and n_{error} the number of packets that arrive with error at the collector node, the reliability R is given by:

$$R = ((n_{pac} - n_{error}) / n_{pac}) \quad (5)$$

For a sensor network be considered energy efficient, the maximum amount of data bits have to be transmitted with the minimum energy consumption. An energy efficiency parameter η may be defined as:

$$\eta = \frac{E_{\min}}{E} \times R \quad (6)$$

where E_{min} is the minimum consumed energy per packet, E is the total consumed energy and R the reliability.

In the simulation model a sensor sends 10000 packets to the sink, considering different amounts of hops and packet sizes. It has been assumed that the CRC code provides perfect error detection. The data can be sent in regular intervals and for instance it may indicate the temperature of an environment or some other variable that could be transmitted with few bytes of data. The data size to be transmitted was chosen to be either 17 or 32 bytes. Although other data sizes could be used, these values may indicate a tendency of the packet behavior. The value of 17 bytes was chosen because is the maximum number of data bytes that the DM1 and BCH packets can transmit. The value of 32 bytes was chosen in order to analyze the behavior of the error control schemes when two packets are necessary to transmit the data bytes (for one time-slot packets). In the simulations with 17 bytes the packets DM3, DH3, DM5 and DH5 are not used because these packets with few bytes would be equal to DM1 or DH1.

IV. Results

A. Number of hops

Figures 4 to 8 show the results obtained for the energy efficiency of each packet as a function of signal-to-noise ratio, for different number of hops (1, 2, 10, 15 and 25). The results are mean values obtained with many simulations. The data size is 17 bytes. For a single hop network (Fig. 4) the AUX1 packet has the best efficiency for SNR values higher than 15 dB, approximately. When the SNR is below this value, BCH3 packet is the best. For 2 hops (Fig. 5) the AUX1 packet still is the best for high SNR, but for approximately 20 dB the adaptive schemes ADP1 and ADP2 have a performance very close to AUX1. For SNR values below 15 dB the BCH2 and BCH3 have the best efficiencies.

With 10 hops (Fig. 6) the relative performance among the packets begins to stabilize. The AUX1 packet only has the highest efficiency for channel conditions above 30 dB. For approximately 30 dB the AUX2 packet becomes the best. The adaptive scheme ADP2 has the best efficiency when the SNR is close to 20 dB and the BCH packet is the best for SNR below 15 dB. It can be noted that when the channel quality is good, it is not necessary a very powerful error correction and the AUX1 and AUX2 packets can be utilized. If the channel conditions are very bad, a code able to correct many errors has to be used, so the BCH packet is the most recommended in such situations. For intermediary conditions, the adaptive schemes ADP1 and ADP2 have the best energy efficiency degree. This behavior of the different error control strategies is approximately the same for 15 and 25 hops (Figures 7 and 8).

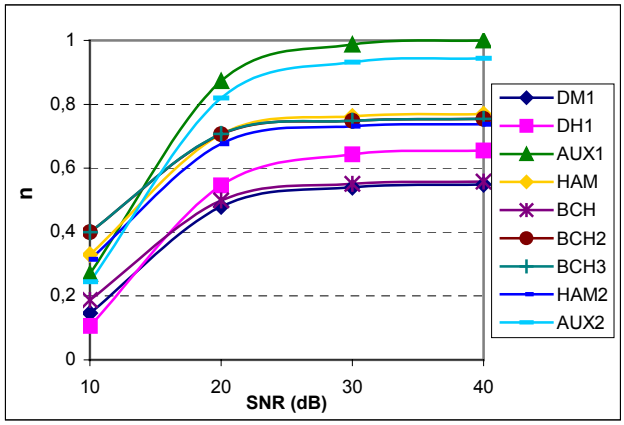

Fig 4. Energy efficiency for 1 hop.

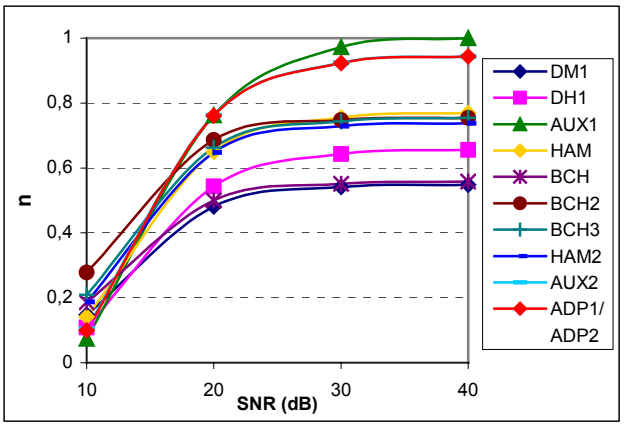

Fig 5. Energy efficiency for 2 hops.

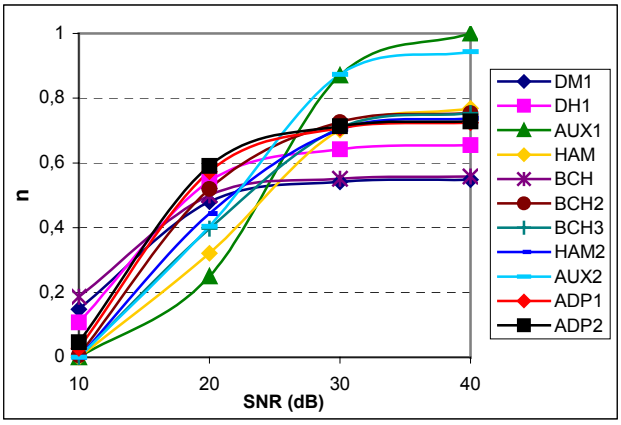

Fig. 6. Energy efficiency for 10 hops.

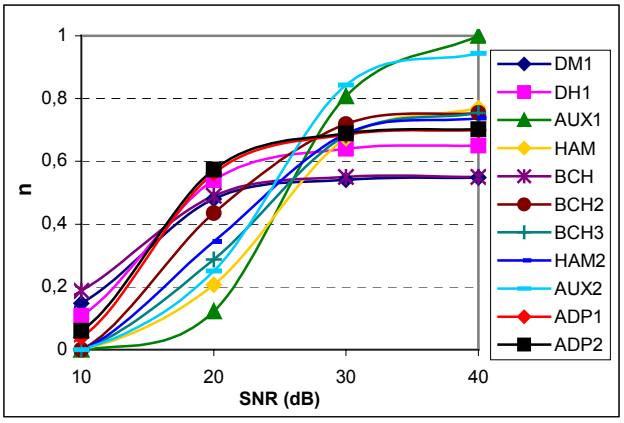

Fig. 7. Energy efficiency for 15 hops.

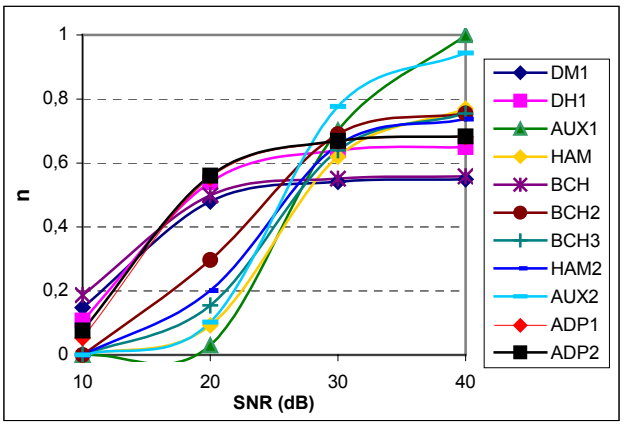

Fig. 8. Energy efficiency for 25 hops.

Figures 9 to 11 show the efficiency η as a function of the number of hops for SNR values of 10, 20 and 30 dB. From these graphs it can be better observed some conclusions taken from Figures 4 to 8. The packet with best energy efficiency for about 30 dB is the AUX1 packet (Fig. 9), for 20 dB the adaptive schemes (Fig. 10) and about 10dB the BCH packet (Fig. 11). The most interesting observation is that the energy efficiency of the packets with retransmission is independent of the number of hops. While the efficiency for the packets without ARQ decrease with both the decrease of the signal-to-noise ratio and the increase of the number of hops, the efficiency of DM1, DH1 and BCH decrease only with the signal-noise ratio. This is an important characteristic of the ARQ Bluetooth packets.

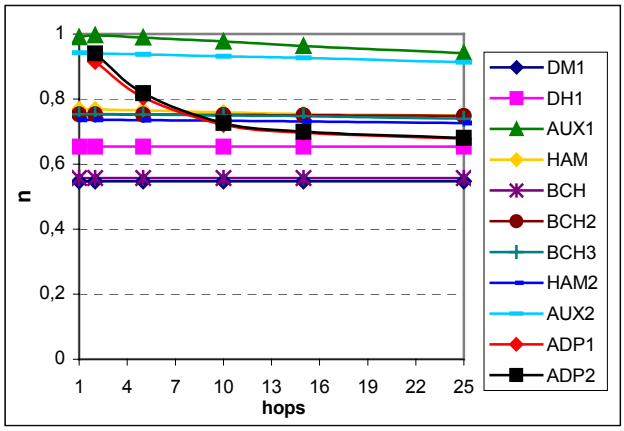

Fig. 9. Energy efficiency for 30 dB.

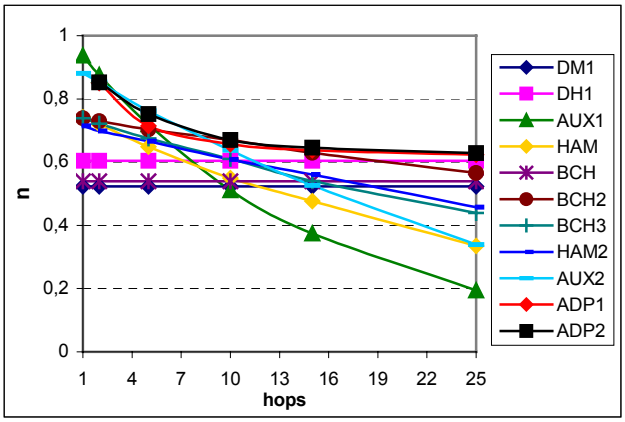

Fig. 10. Energy efficiency for 20 dB.

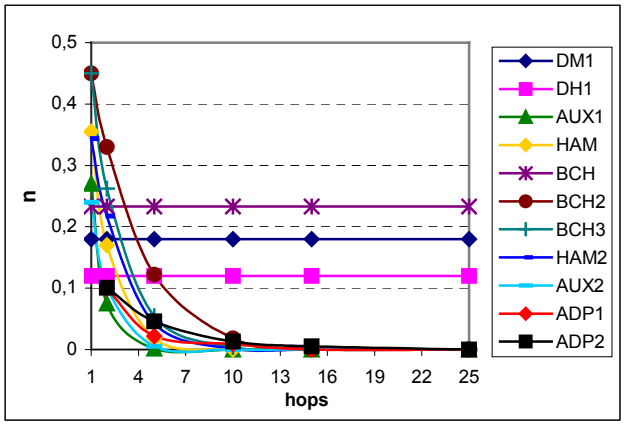

Fig. 11. Energy efficiency for 10 dB.

B. *Data Size*

In this Subsection it is shown the results obtained for a 32 bytes data packet. For the AUX1 packet, first it is transmitted one packet with 29 data bytes (the maximum of AUX1 packet) and then a second packet with 3 data bytes (resulting in 32 data bytes).

Figures 12 and 13 show the energy efficiency as a function of the signal-to-noise ratio for 2 and 15 hops with a 32 bytes data packet, respectively. It was used only one adaptive scheme, the ADP1 (replacing the DH1 packet with a DH3 packet). For 2 hops, the ADP1 scheme has the best efficiency for SNR values approximately between 15 and 30 dB. Above 30 dB the best packet is the AUX2 and below 15 dB the DM3 packet is the best packet. For 15 hops, the ADP1 scheme has a performance very close to DH3. For SNR values about 10 dB, the DM3 packet is the most energy efficient scheme.

In the Bluetooth case, data size higher than 32 bytes would not benefit from the energy savings that the AUX1 packet and its custom coding strategies may present, leaving the choice of the packet between DM3 and DH3 or DM5 and DH5. This occurs because while the AUX1 packet and the other one timer-slot packets (see Table I and II) need more than one packet to send the data (adding extra header and access code), the DH3 and DM3 packets carry all information in just one packet.

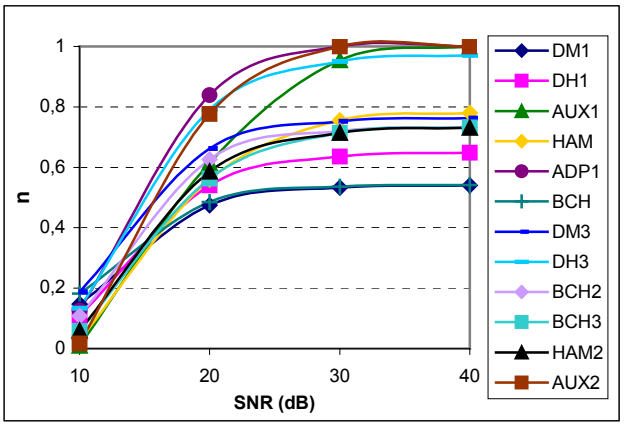

Fig. 12. Energy efficiency for 2 hops, 32 bytes.

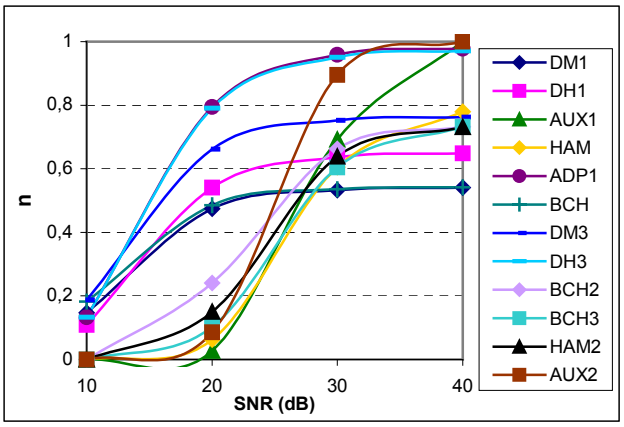

Fig. 13. Energy efficiency for 15 hops, 32 bytes.

V. Conclusion

The energy efficiency metric proposed in this paper considers in just one parameter the energy and reliability constraints of wireless sensor networks. New Bluetooth packet types were introduced using custom error control schemes in the AUX1 packet. These modifications include a CRC for error detection (without ARQ), BCH code with and without CRC and Hamming code with and without CRC. Two adaptive error control schemes were proposed, ADP1 and ADP2, that change the error control strategy accordingly to the number of hops that the packet traveled through the sensor network. The results have shown that for high values of SNR the packets with little or none error protection present the best energy efficiency. For a network with 1 or 2 hops and low SNR the best packet is the BCH for 17 data bytes and DM3 for 32 data bytes. When the network has more hops and low SNR the BCH packet is the most efficient, because of its ability to correct more errors, despite of more energy consumption. In intermediary situations (about 20 dB) the adaptive schemes have the best performance. Although the simulation model and the error control schemes presented in this paper were applied to the Bluetooth standard, they can be well adapted to other wireless technologies for sensor networks.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A survey on sensor networks", *IEEE Communications Magazine*, pp. 102-114, August 2002.

[2] Bluetooth SIG, "Specifications of the Bluetooth system", *Core Version 1.2*, Novembro 2003. http://www.bluetooth.com

[3] O. Kasten and M. Langheinrich, "First experiences with Bluetooth in the smart-its distributed sensor network", Workshop on Ubiquitous Computing and Communications, Barcelona, Spain, September 2001.

[4] M. Leopold, M. D. Dydensborg and P. Bonnet, "Bluetooth and sensor networks: a reality check", 1st ACM Conference on Sensor Systems, Los Angeles, CA, USA, November 2003.

[5] V. Mehta and M. El Zarki, "Bluetooth based sensor network for civil infrastructure health monitoring", *Wireless Networks*, Kluwer Academic Publishers, vol. 10, pp. 401-412, July 2004.

[6] S. Saginbekov and I. Korpeoglu, "An energy efficient scatternet formation algorithm for Bluetooth-based sensor networks", 2nd European Workshop on Wireless Sensor Networks, Istanbul, Turkey, February 2005.

[7] H. Mathias, D. Jan and T. Dirk, "Energy-efficient data collection for Bluetooth-based sensor networks", IEEE Instrumentation and Measurement Technology Conference, Italy, May 2004.

[8] J. H. Kleinschmidt, M. E. Pellenz and L. A. P. Lima Jr., "Uma aplicação de redes de sensores usando Bluetooth", XXI Brazilian Telecommunications Symposium, Belém, Brazil, September 2004 (*in Portuguese*).

[9] Y. Sankarasubramanian, I. F. Akyildiz and S. W. Mc Laughlin, "Energy efficiency based packet size optimization in wireless sensor networks", Proc. of Sensor Network Protocols and Applications, 2003.

[10] H. Karvonen, Z. Shelby and C. Pomala-Ráez. "Coding for energy efficient wireless embedded networks", International Workshop on Wireless Ad-hoc Networks, 2004.

[11] J. Meer, M. Nijdam and M. Bijl, "Adaptive error control in a wireless sensor network using packet importance valuation", Hardware/software co-design, Enschede, Netherlands, May 2003.

[12] J. H. Kleinschmidt, W. C. Borelli and M. E. Pellenz, "Power efficient error control for Bluetooth-based sensor networks", IEEE Local Computer Networks Conference, Sydney, Australia, November 2005.

[13] M. C. Valenti and M. Robert, "Custom coding, adaptive rate control and distributed detection for Bluetooth", Proc. IEEE Vehicular Technology Conference, Vancouver, BC, September 2002.

[14] C. Desset and A. Fort, "Selection of channel coding for low-power wireless systems", Proc. IEEE Vehicular Technology Conference, Jeju, Korea, April 2003.