Power Efficient Error Control for Bluetooth-based Sensor Networks

João H. Kleinschmidt, Walter C. Borelli

School of Electrical and Computer Engineering
State University of Campinas - UNICAMP
{joaohk, borelli}@dt.fee.unicamp.br

Marcelo E. Pellenz

Graduate Program in Computer Science
Pontifical Catholic University of Paraná - PUCPR
marcelo@ppgia.pucpr.br

Abstract

This paper studies different error control schemes in wireless sensor networks with Bluetooth technology. The tradeoff between reliability and energy consumption of Bluetooth packets are analyzed, using different error control techniques, such as retransmission and channel coding The AUX1 packet is utilized for custom coding and also are proposed adaptive techniques based on the number of hops of the network. The wireless channel is modeled with Rayleigh fading. The results obtained may be used as references to determine the packet type in a sensor application.

1. 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]. Some of the applications of sensor networks are monitoring disaster areas, managing inventory, monitoring product quality, environmental observation, monitoring of human physiological data and so on. The network protocols, such as formation algorithms, routing and management, must have self-organizing capabilities. In general, sensor networks differ from ad hoc networks in some aspects [1], namely, the number of sensor nodes can be very high; sensor nodes are prone to failures; sensor nodes are densely deployed; the topology of the network can change frequently; sensor nodes are limited in computational capacities, memory and energy.

Topics like the development of new medium access control (MAC) protocols for sensor networks and how to use the existing ones, have received great attention recently. The MAC protocol is responsible for the creation of the network infrastructure and to share communication resources between the nodes.

Bluetooth [2] is a low cost wireless radio technology designed to eliminate wires and cables between mobile and fixed devices over short distances, allowing the formation of ad hoc networks. The lower layers of Bluetooth became the IEEE 802.15 standard for wireless personal area networks (WPANs). 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 devices can communicate with each other forming a network (called piconet) with up to eight nodes. Within a piconet, one device is assigned as a master node and the others act as slave nodes. Devices in different piconets can communicate using a structure called scatternet. The channel is divided in time slots of 625 μs. A time-division duplex (TDD) scheme is used 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 applications, together with its low cost, multi-hop capabilities, device discovery process and energy saving modes. In [3] and [4] sensor networks were implemented using Bluetooth as the MAC protocol. Some of the important issues are the scatternet formation and routing in order to achieve the multi-hop communication. Although many protocols have been proposed in the literature, few of them consider sensor applications, which have hard energy requirements. Some proposals were presented in [5], [6] and [7].

The wireless channels can have high bit error rates due to interference and the multipath propagation that characterizes the radio channel, leading to energy waste. In order to improve the reliability of the data sent in the

Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)
0-7695-2421-4/05 $20.00 © 2005 IEEE

wireless channel, many techniques can be employed, such as automatic repeat request (ARQ), forward error correction (FEC) or transmission power control.

In this paper we analyze the performance of Bluetooth data packets in terms of energy consumption and reliability. Some previous works analyze the performance of the packets [8], but consider the throughput as the performance metric, that is a secondary parameter in sensor networks, that have low data rates. The Bluetooth packets used in data transmission have different types of error correction, as retransmission strategy or Hamming codes. The structure of these standard packets is briefly presented in Section 2. It is also discussed custom coding [9] in Bluetooth and is proposed an adaptive scheme of error control. In Section 3 is showed the simulation model. Section 4 presents the results obtained and Section 5 gives the final considerations.

2. Bluetooth error control strategies

The Bluetooth specification [2] defines seven data packets (asynchronous). 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 the packet type, flow control and acknowledgement. The access code is error robust, because the 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 (Cyclic Redundancy Check) code. The receiver indicates in the next return packet if 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 the single bit errors and detects all two bits errors in a code word. Table 1 shows this information for each asynchronous packet.

Table 1. Asynchronous packet types

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

A received packet is not accepted when anyone of the five events 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. In [8], the probabilities of these events were derived and are adapted to be used in this paper.

The synchronization is made correlating the demodulator output with a stored copy of the access code. A packet is synchronized if the correlator output exceeds a given threshold T. The frame is synchronized if at least T of the 72 bits of the access code were properly demodulated ($T = 65$ in this work). The synchronization with the received packet occurs if there are no more than (72 – T) errors in the received access code:

$$P[\overline{A}] = \sum_{k=0}^{72-T} \binom{72}{k} \cdot [p(\gamma_f)]^k \cdot [1 - p(\gamma_f)]^{72-k}, \quad (1)$$

where $p(\gamma_f)$ is the symbol error probability of the forward channel as a function of the average received signal-to-noise ratio (SNR) $\bar{\gamma}$. Since the return packet also has an access code of 72 bits, the probability for the event D has the same form of event A,

$$P[\overline{D}] = \sum_{k=0}^{72-T} \binom{72}{k} \cdot [p(\gamma_r)]^k \cdot [1 - p(\gamma_r)]^{72-k}, \quad (2)$$

where $p(\gamma_r)$ is the symbol error probability of the reverse channel. The forward channel is used to send data packets and the reverse channel indicates the success or not of the transmission of a packet (for unidirectional transmission). The events B or E occur if any of the eight triples of the repetition code (3,1) were incorrectly decoded,

$$P[\overline{B}] = \left\{3p(\gamma_f)[1 - p(\gamma_f)]^2 + [1 - p(\gamma_f)]^3\right\}^{18} \quad (3)$$

$$P[\overline{E}] = \left\{3p(\gamma_r)[1 - p(\gamma_r)]^2 + [1 - p(\gamma_r)]^3\right\}^{18} \quad (4)$$

The most probable error is that defined by event C. For DHx packets it occurs when any of the data bytes were received with error:

$$P[\overline{C}] = [1 - p(\gamma_f)]^b, \quad (5)$$

where b is the size of the payload in bits. For DMx packets the data are protected by a Hamming code, where B is the number of blocks with 10 bits. The probability of event C for the DMx packets is:

$$P[\overline{C}] = \left[15p(\gamma_f)[1 - p(\gamma_f)]^{14} + [1 - p(\gamma_f)]^{15}\right]^B \quad (6)$$

Bluetooth uses GFSK modulation with time-bandwidth product $BT=0.5$ and modulation index between 0.28 and 0.35. The error symbol probability

$p(\gamma)$ for the GFSK modulation must be applied in equations (1) and (6) and is given by [11]:

$$p(\gamma) = Q_1(a,b) - \frac{1}{2} e^{(a^2+b^2)/2} I_o(ab) \quad (7)$$

Q_1 *(a,b)* is the Q-Marcum function, I_o is the modified Bessel function of first kind and *a* e *b* are constants that depend on the signal-to-noise ratio [11]. Thus, the packet error probability of the forward channel, PER_f, and reverse, PER_r, can be defined [12] as:

$$PER_f = 1 - \int_0^\infty f(\gamma_f) P[\overline{A}] P[\overline{B}] P[\overline{C}] \, d\gamma_f \quad (8)$$

$$PER_r = 1 - \int_0^\infty f(\gamma_r) P[\overline{D}] P[\overline{E}] \, d\gamma_r \quad (9)$$

where $f(\gamma_f)$ and $f(\gamma_r)$ are the probability density functions of the forward and reverse channels, respectively.

The packets defined by the standard have specific error control, but custom coding can be implemented using the AUX1 packets [9]. Using this packet, the ARQ is turned off and the Bluetooth device delivers the received bits independently if they are correct or not. While the six asynchronous packets with ARQ maintain a reliable link with random delay (which approaches infinity for low values of SNR), the AUX1 packet provides an unreliable link with delay of one time slot. The remaining packets cannot be used to implement other error correcting codes without modification in the specification [2]. In [9] is proposed the use of BCH codes with the CRC code for error detection. In this case, as the ARQ is turned off in the device, it must be implemented at the application layer. The coder is implemented inserting in the payload of the AUX1 packet a (232, k) BCH code. 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. Only the payload decoding probability (event C) is different, and is given by equation (10). The code considered in this paper was a (232, 156) binary BCH code that can correct up to *t*=10 errors [9].

$$P[\overline{C}] = \sum_{k=0}^{t} \binom{232}{k} \cdot \left[p(\gamma_f)\right]^k \cdot \left[1 - p(\gamma_f)\right]^{232-k} \quad (10)$$

We propose other modification in the AUX1 packet: apply in the payload the same Hamming code of the DMx packets, but without the use of CRC. Although this strategy can decrease the reliability of transmitted packets, in terms of energy consumption can be very useful, because is not necessary to send a return packet to indicate the success of the transmission. Table 2 shows the information of error control for the two new packet types.

Using the same error control scheme for the whole network can be a good choice for some cases, but not always. In some situations if a packet gets lost is not a big problem, but sometimes the loss of a packet cannot be tolerated.

Table 2. Packet types with custom control coding

Packet	Time slots	Payload (bytes)	FEC	CRC and ARQ
HAM	1	0-18	Hamming (15,10)	No
BCH	1	0-17	BCH (232,156)	Yes

An adaptive scheme can be used to verify the importance of a packet and choose an efficient error control for that particular packet. In Bluetooth case, to change the error correction signifies 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 an approach based on [10].

The importance of a packet is determined using the multi-hop principle, as show in Fig. 1. The packet utilized and consequently, the error control technique applied, will be based on the number of hops that the packet had in the network. The sensor node sends a data packet containing the information of the temperature of an environment, for example, to the collector node. The collector is a node that receives the data of all the nodes of the network or of some area of the network.

However, before the packet reaches the collector node, it must flow through the other nodes of the network that can be sensors or another type of node with routing capacity. If the packet gets lost at the first hop, only the energy to send the packet from sensor to node 1 was lost. If the packet is corrupted after node 4 sends it to the collector, more energy was spent to send the packet through the network. Thus, a packet is more important as more hops it had in the network. An adaptive scheme can use stronger error control techniques for packets that have more hops and less error control for the first hops.

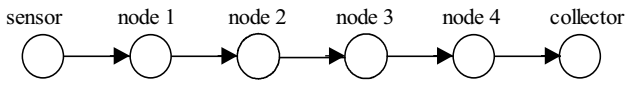

Figure 1. Example of a multi-hop sensor network

3. Simulation model

In order to evaluate the energy consumption of the packets, simulations were implemented using the Matlab software and are described in the sequence. The network considered is showed in Fig. 1, where a sensor must send

data to the collector node. This structure is only one of many routes of the sensor network that can have different topologies. Fig. 1 represents one path of many that can exist in the sensor network. It is being assumed that the Bluetooth scatternet was formed and that the scheduling policy and the routes are also defined, using protocols as the proposed in [5], [6] and [7]. The packet data is generated by the sensor node, that sends it to node 1, and so on, until it reaches the collector node. The wireless channel was modeled using the Rayleigh fading, whose probability density function is given by:

$$f(\gamma) = \frac{1}{\bar{\gamma}} \exp\left(-\frac{\gamma}{\bar{\gamma}}\right), \quad \text{for} \quad \gamma \geq 0 \tag{11}$$

where $\bar{\gamma}$ is the average received SNR and γ is the instantaneous SNR.

Using equation (11) in (8) and (9) we calculate the error probabilities of each packet. These probabilities are given as a function of the signal-to-noise ratio. When a node receives a packet it is verified if errors occurred in the reception. If there were no errors the packet is sent o to the next node. In the packets with ARQ, a packet is sent to the transmitter indicating the success of the transmission (a Bluetooth NULL packet). If errors are detected, the packet is discarded (in the case of packets without ARQ) or is sent to the transmitter a NULL packet indicating the unsuccessful transmission, so the packet will be sent again. It is important to note that the NULL packet used to acknowledge a transmission can also be corrupted, although do not carry any data (it has only the fields of access code and header).

For the simulation of the energy consumption was used a simplified model. 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, which are the tasks with more energy consumption. The energy to code or decode a packet was not considered. The coding is generally simpler and has less consumption than decoding [10][13]. It is also considered that the Bluetooth device is in the connected state. Energy savings can be made during idle times using the power management schemes of Bluetooth (hold, park and sniff). Considering that the reception of a determined number of bits consume approximately 75 per cent of the energy spent to transmit the same number of bits, as stated in [10], the total energy consumed E is given by:

$$E = n_{bits}*T*1 + n_{bits}*R*0.75 + n_{ack}*T_a*1 + n_{ack}*R_a*0.75 \tag{12}$$

being n_{bits} the total number of bits of a packet (access code, header and payload), T the total number of transmitted packets (including retransmissions), R the total number of received packets (including retransmissions), n_{ack} 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 that arrive correct at the collector 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 C is given by:

$$C = ((n_{pac} - n_{error})/ n_{pac}) * 100 \tag{13}$$

For the adaptive error control, each packet must have a counter with the number of hops that packet traversed 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 less error control is used in the first hop and a packet with more powerful coding in the others. For the case with two hops, the first scheme (ADP1) uses the AUX1 packet in the half of the transmissions of the first hop and BCH in the second hop. In the second adaptive scheme the AUX1 packet is used in the first hop and DH1 in the second hop. For the five hops case, ADP1 uses AUX1 in the first hop and BCH in the next hops. In the second scheme ADP2, the first hop uses AUX1, the second hop HAM and BCH for the others.

In the simulation the sensor sends 10000 packets to the collector, with four different channel conditions. The first simulation is an error free environment. In other cases the channel is modeled with Rayleigh fading with 30dB, 20dB and 10dB of average received signal-to-noise ratio. The data size to be transmitted has 17 or 32 bytes. The data can be sent in regular intervals and can indicate the temperature of an environment or other variable that can be transmitted with few bytes. The value of 17 bytes was chosen because is the maximum number of data bytes that the DM1 and BCH packets can transmit.

In order to transmit 32 information bytes, two packets are needed for the DM1, DH1, AUX1, HAM and BCH. While the first packet has the maximum number of data bytes, the second contains only the number of bytes to complete the 32 data bytes. In the simulations with 17 bytes the packets DM3, DH3, DM5 and DH5 are not used because these packets with few bytes would be reduced to DM1 or DH1. With 32 bytes the DM3 and DH3 packets are used and need only one packet to send the data.

4. Results

Fig. 2 to 12 show the results obtained of energy consumption and reliability of received packets. In the packets with retransmission is being considered that the CRC code detects all errors. It is for this reason that these packets always reach 100 per cent of reliability, at the

cost of a higher energy consumed. The results are mean values obtained with several simulations.

For a channel without errors (Fig. 2 and 9), the AUX1 packet always has advantages in comparison with others, independently of the number of hops or data size, because it reaches the same 100 per cent of reliability with the less energy consumption. In this condition, the worst packet is the DM1, because the redundancy inserted by the Hamming code and the return packet adds bits to be transmitted without necessity. However, a free-error channel does not occur in practice. For a Rayleigh channel with 30 dB of signal-to-noise ratio (Fig. 3, 6 and 10) some errors begin to appear in the transmitted bits. Again the AUX1 packet has the least consumption, but the reliability decreases. Even though it is not 100 per cent, it is very high, usually above 90 per cent. This loss is not always a problem, because in many applications several sensors collect redundant information of an environment. If some data get lost, the remaining sensors can still give reliable measures. The HAM packet is more reliable, but also with higher energy consumption. The adaptive scheme 2 had a good efficiency for the 2-hop case (Fig.3 and 10), consuming just a little more energy than the no ARQ packets and reached almost 100 per cent of reliability.

In channels with 20dB of SNR (Fig. 4, 7 and 11), several changes can be noted. Since the channel conditions are not good, with more errors, the packets with error control begin to have a better performance. The AUX1 packet still reaches almost 80 per cent of reliability for 2 hops and 17 bytes (Fig. 4), but with 5 hops (Fig. 7) the reliability decreases considerably, because the error probability is accumulated on each hop. The adaptive schemes ADP1 and ADP2 have a good improvement at these conditions because the use of more efficient error control contains the accumulation of error probability. The packets with retransmission have a significant improvement in the energy consumption if compared with 30dB. Another important observation is that for packets with 32 data bytes (Fig. 9, 10, 11 and 12), the DH3 packet has high reliability and consumption almost equal to the no ARQ packets. This occurs because while the others need two packets to send the data (adding one more header and access code), the DH3 (and DM3) send all information in only one packet. It already can be noted that increasing the number of data bytes, these packets always will have advantage compared to the others. For higher data-rate sensor networks the packets with custom coding and the adaptive schemes will not have good efficiency, because more packets are needed to send the same amount of data.

When the signal-to-noise ratio is only 10dB (Fig. 5, 8 and 12), the channel will have many errors and the energy consumption have a great increase for the packets with ARQ, due to the great number of retransmissions. At the same time, the packets without ARQ have its reliability almost reduced to zero. From all of the ARQ packets, the BCH is that have the least energy consumption for packets with 17 bytes (Fig. 5 and 8) and for 32 bytes (Fig. 12) the DM3 packet. As it expected, coding gives better results when the channel conditions are worse. The adaptive schemes give a little improvement at the energy consumption, but the reliability is not high.

From the results obtained can be noted that for each channel condition a different packet has a better performance and a method to estimate the signal-to-noise ratio would be very useful. However, this requires that the device has this characteristic and estimates the SNR for each received packet. Each application can have different requirements and if the reliability demanded is not very high, a lot of energy can be saved. The number of hops also directly affects the choice of the packet. For the Bluetooth case, if the data size is more than 32 bytes it is not interesting to use the AUX1 packet or the HAM and BCH, leaving the choice of the packet between DM3 and DH3 or DM5 and DH5.

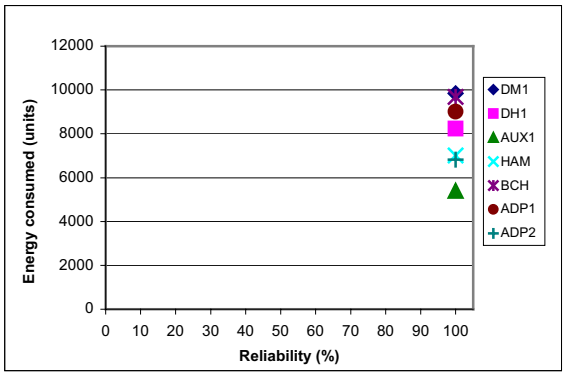

Figure 2. Error-free, 17 bytes, 2 hops

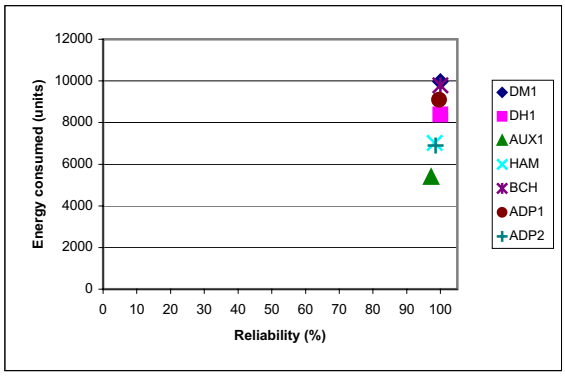

Figure 3. Rayleigh 30 dB, 17 bytes, 2 hops

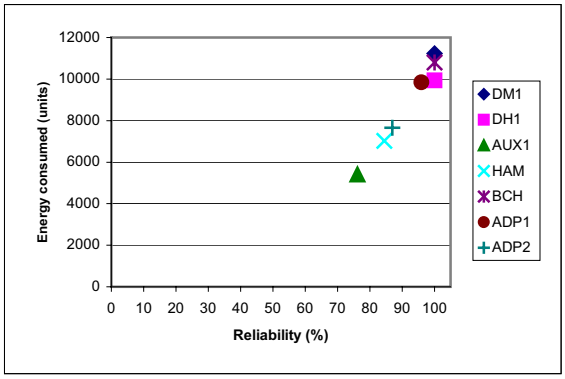

Figure 4. Rayleigh 20 dB, 17 bytes, 2 hops

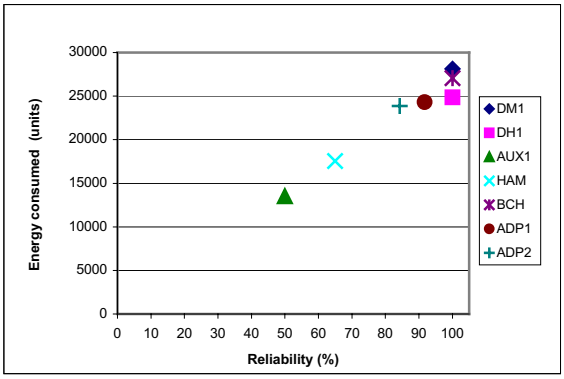

Figure 7. Rayleigh 20 dB, 17 bytes, 5 hops

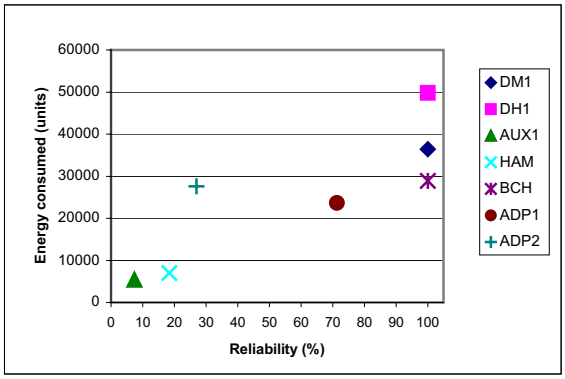

Figure 5. Rayleigh 10 dB, 17 bytes, 2 hops

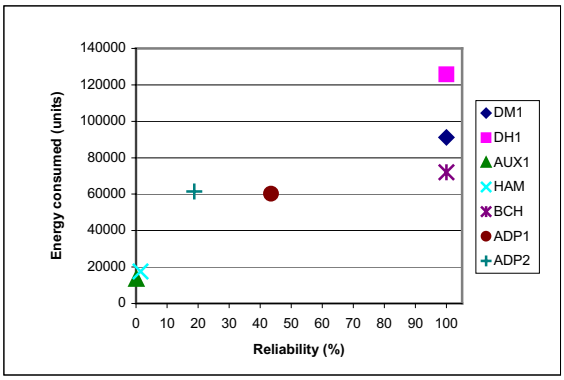

Figure 8. Rayleigh 10 dB, 17 bytes, 5 hops

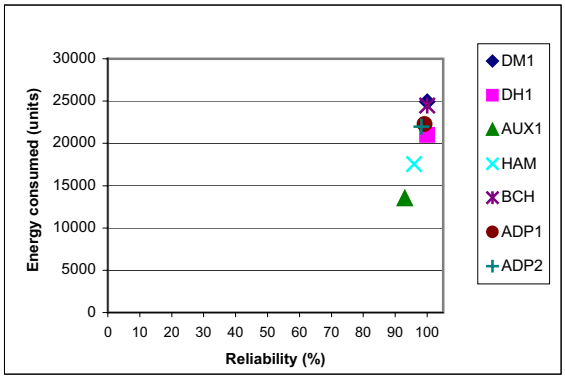

Figure 6. Rayleigh 30 dB, 17 bytes, 5 hops

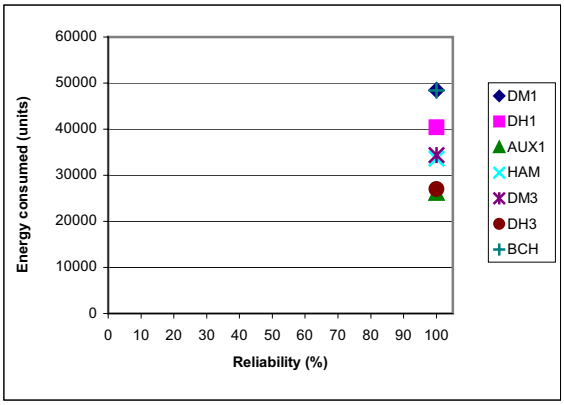

Figure 9. Error-free, 32 bytes, 2 hops

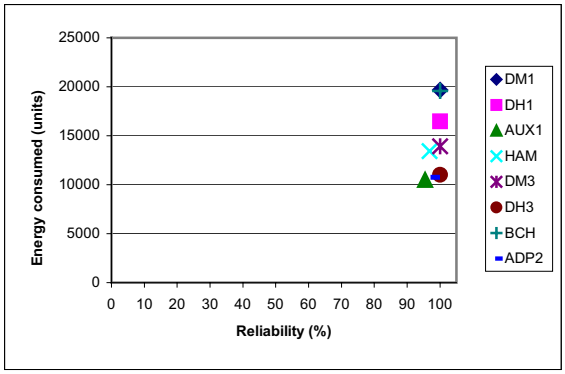

Figure 10. Rayleigh 30 dB, 32 bytes, 2 hops

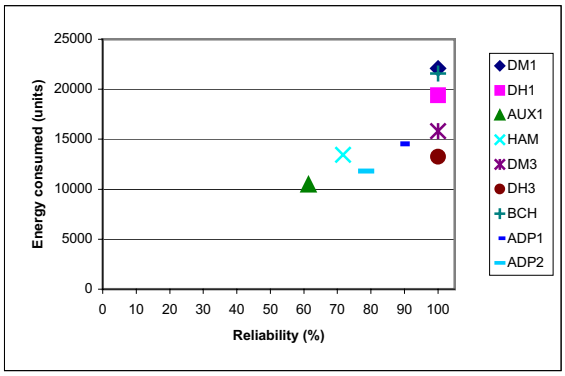

Figure 11. Rayleigh 20 dB, 32 bytes, 2 hops

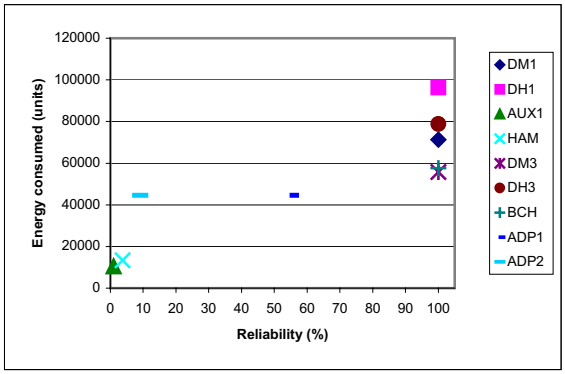

Figure 12. Rayleigh 10 dB, 32 bytes, 2 hops

5. Conclusion

In this paper was studied the issue of energy consumption in Bluetooth sensor networks using different techniques of error control, based on retransmission and channel coding. The AUX1 packet can be useful in sensor networks because it do not use ARQ and permits that specific strategies of error control be employed. The utilization of BCH code and Hamming code without CRC was proposed using the AUX1 packet. Adaptive schemes ADP1 and ADP2 that change packet type accordingly to the number of hops were also proposed. Other similar adaptive techniques can be proposed to decrease the energy consumption in bad channel conditions. In a channel with good conditions the packets without ARQ have low consumption and high reliability. Although the energy spent to code and decode was not considered, the AUX1 packets and the adaptive schemes cam have additional benefits, because have less coding and will spent less energy. The results obtained can serve as indications of the packet type to be used in a sensor application for a given reliability.

6. 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*, November 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] M.C. Valenti, M. Robert and J.H. Reed, "On the throughput of Bluetooth data transmissions", IEEE Wireless Communications and Networking Conference, Orlando, USA, March 2002.

[9] 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.

[10] 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.

[11] J. Proakis, Digital Communications, New York, NY: McGraw-Hill, 4th edition, 2001.

[12] J.H. Kleinschmidt, M.E. Pellenz and L.A.P. Lima Jr., "Evaluating and improving Bluetooth piconet performance over Nakagami-*m* fading channels", Proc. of the Ninth IEEE International Symposium on Computers and Communications, Alexandria, Egypt, July 2004.
[13] C. Desset and A. Fort, "Selection of channel coding for low-power wireless systems", Proc. IEEE Vehicular Technology Conference, Jeju, Korea, April 2003.

Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)
0-7695-2421-4/05 $20.00 © 2005 IEEE