Sensors & Transducers, Vol. 189, Issue 6, June 2015, pp. 22-30

Sensors & Transducers

http://www.sensorsportal.com

Energy Efficient Payload Aggregation in WSNs

Ákos MILÁNKOVICH, Gergely ILL, Károly LENDVAI, Sándor IMRE, Sándor SZABÓ
Department of Networked Systems and Services, Budapest University of Technology and Economics, Magyar Tudosok krt 2, Budapest, H-1117, Hungary
E-mail: milankovich@hit.bme.hu, ill@hit.bme.hu, lendvai@hit.bme.hu, imre@hit.bme.hu, szabos@hit.bme.hu

Received: 29 May 2015 /Accepted: 26 June 2015 /Published: 30 June 2015

Abstract: Creating wireless sensor networks requires a different approach than traditional communication networks because energy efficiency plays a key role in sensor networks, which consist of devices without external power. The amount of energy used determines the lifetime of these devices. In most cases data packets are less sensitive to delay, thus can be aggregated, making it possible to gather more useful information reducing the energy required to transmit information. This article discusses the energy efficiency of different Forward Error Correction algorithms and presents a method to calculate the optimal amount of aggregation of the data packets in terms of power consumption, while taking into account the Bit Error Rate characteristics of the wireless channel. The contribution of this paper is a general method to improve the energy efficiency of wireless sensor networks by using the optimal amount of aggregation in case of different Forward Error Correction codes and channel characteristics. The presented results can be applied to any packet-based wireless protocol.

Keywords: Wireless sensor networks, Aggregation, Energy efficiency, FEC, BER.

1. Introduction

The use of wireless sensor networks is becoming popular in various areas such as production, environment and healthcare monitoring, smart metering, intelligent home, precision agriculture, etc. During the design and implementation of such systems, special attention should be paid to the energy consumption of the network nodes, as they usually operate on battery power. Moreover, in many applications, it is possible that the nodes transmit the useful information in an application-specific predefined time T delay instead of real-time communication. Such systems are called Delay-Tolerant Networks (DTN).

This paper focuses on the energy consumption of sensor networks with the restrictions defined by the operation of DTNs. Our goal is to minimize the energy consumption of network nodes, taking into account the BER (Bit Error Ratio) quality of the radio channel to maximize battery life. This paper aims to reach this goal by the means of the following technique: using aggregation and (Forward Error Correction) FEC codes. This method is applied in the ISO-OSI Physical and Data link layers. The optimal aggregation number is determined to decrease the amount of consumed energy. The method was developed for multi-hop wireless sensor networks with stationary nodes.

This paper is an extended version of [1]. The paper is organized as follows: Section 2 and Section 3 introduce the system model along with the considered parameters of the sensor network hardware and communication protocol. Section 4 describes the method of using aggregation to increase efficiency. Finally, Section 5 concludes the paper.

1.1. Related Work

Various optimization problems in wireless sensor networks were extensively covered by the literature. In this section we collect the most important papers dealing with some aspects of energy efficiency.

The ideal packet size is calculated in papers like [2-3]. The relations of SNR, BER and the used modulation on the radio channel are presented in [4-5]. Energy efficiency of routing protocols are discussed in [6-7]. The advantages of clustering algorithms are showed in [8]. In paper [9-10], FEC schemes are evaluated for multi-hop communication. The benefits of packet aggregation are also investigated in [11-13].

In our previous works [14-15], we presented an optimization method for determining the ideal size of an aggregated packet according to the channel characteristics and we extended that study when using FEC. In this paper we expand our previous work and determine the ratio of energy usage in case of aggregation and without it, considering the packet losses and corruptions on the radio channel. Moreover we investigate effects of using FEC for these scenarios.

2. Description of the System Model

The goal during communication is, considering the constraints (e.g. the information has to arrive within time interval T) to transmit the payload bits over the wireless channel with the least possible energy consumption. The data packets are structured according to Fig. 1.

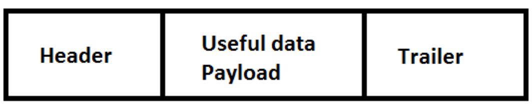

Fig. 1. Packet structure.

The header and trailer are considered to have fixed length, which are determined by the applied communication protocol, the types of encryption and error correction code. From the point of transmitted data, these are not considered useful information, but overhead. The overall length of the header and trailer is ω bits.

The useful data consists of fix, predetermined length of elements and structure. The size of this payload data is φ bits. To maximize the energy efficiency of the system, the useful bits/all transmitted bits ratio has to be maximized. According to this goal and assuming no error in the transmission the most possible useful data can be transmitted in one packet, which means, that aggregation of the information into one packet is necessary, because this guarantees that the overhead ratio in the packet will be minimal. In a data packet n pieces of data elements of φ bits length are transmitted, so the useful data amount is n times φ bits.

In a real world scenario, transmission without errors in the channel is impossible. The communication can be achieved only with a certain amount of bit error rate. In this case, the pervious statement, that the lengthiest packet is the most energy efficient is not true, because the longer the packet, the more likely it will suffer error during transmission and hence it has to be resent. Error correction coding can help to recover some of the corrupted bits.

The following calculations can be carried out to any other hardware. The formulas are considered general solutions. The described protocol is developed by the authors for delay-tolerant data transfer, but the only parameters considered are the amount of overhead and the payload length and whether ACK is needed for the communication. Having the knowledge of these parameters the formulas can be applied for other protocols. The parameters of the aforementioned devices were determined using their datasheets.

To determine the particular size of the parts of the packet, the calculations are based on protocol developed by the authors for wireless sensor networks. The communication protocol differentiates two packet classes. One is responsible for network management (e.g. discovery), the other for data communication. The latter category has two message types. One is the data packet itself, and the other is the corresponding acknowledgement (ACK). The transmission is successful, if the packet was sent and the ACK is received. If any of the packets suffers bit error during transmission, it has to be resent because there is no error correction coding. Therefore the calculations can be simplified. The ACK message does not hold useful bits regarding the information to be transmitted, so it is calculated as overhead. Therefore, we add the length of ACK to the packet length. The ACK message is the same as the header part of a traditional data packet, which means its size is 18 bytes. During optimization we do not take management messages into account, because we cannot influence their packet size.

3. Constants and Parameters

In this section we introduce the parameters shared by both of the energy-saving solutions. The parameters and their values are summarized in Table 1. The demo system consists of an Atmel AVR XMEGA A3 microcontroller [16] and a TI CC1101 433 MHz radio module [17]. Both devices are extremely suitable for sensor networks, due to their low power consumption, reliability and low price.

The symbols and parameters used in this article are: B : 9.6 kbaud/sec. Using GFSK modulation, one symbol carries one bit, which equals 9.6 kbit/sec.

Table 1. Common parameters for calculations.

Symbol	Description	Value	Unit
$\boldsymbol{\omega_h}$	Length of header	128	bit
$\boldsymbol{\omega_{MAC}}$	Length of MAC	16	bit
$\boldsymbol{B}$	Transfer rate	9600	bit/s
$\boldsymbol{n}$	Aggregation number	1-100	pcs
$\boldsymbol{\varphi}$	Length of payload	80	bit
$\boldsymbol{BER}$	Bit error rate	$4 \cdot 10^{-3}$, $4 \cdot 10^{-4}$, $4 \cdot 10^{-5}$	$probability$
$\boldsymbol{N}$	Block size of FEC	depends on FEC	bit
$\boldsymbol{K}$	Code length of FEC	depends on FEC	bit
$\boldsymbol{t}$	Error correcting capability of FEC	depends on FEC	bit
$\boldsymbol{r}$	Number of retransmissions	depends on FEC and BER	pcs
$\boldsymbol{i_{rx}}$	RX current	35.7	mA
$\boldsymbol{i_{tx}}$	TX current	55.7	mA
$\boldsymbol{T_{tst}}$	Time needed for RX-TX state change	799	μs
$\boldsymbol{T_{rst}}$	Time needed for TX-RX state change	799	μs
$\boldsymbol{T_{1enc}}$	Time needed for encoding 1 bit	0.0916	$\mu s/bit$
$\boldsymbol{T_{1dec}}$	Time needed for decoding 1 bit	0.0916	$\mu s/bit$
$\boldsymbol{u}$	Voltage	3	V
$\boldsymbol{i_{active}}$	Atmel CPU active	15.7	mA
$\boldsymbol{i_{tst}/i_{rst}}$	Current needed for RX-TX state change	24.1	mA
$\boldsymbol{i_{enc}/i_{dec}}$	Current needed by AES coder and µcontroller during coding and decoding	15.92	mA
$\boldsymbol{f_{CPU}}$	Atmel CPU clock speed	32	MHz

i_{tx}: 40 mA (at +10 dBm output power). This value should be increased by the 15.7 mA current draw of the microcontroller, but in case of transmission, the microcontroller encodes simultaneously, so this value is considered in I_{enc}. ([17] page 9, Table 4).

i_{rx}: 20 mA (at sensitivity limit). This value should be increased by the 15.7 mA current draw of the microcontroller, but similarly as the transmission, in case of receiving, the microcontroller simultaneously decodes, so this value is considered in I_{dec}. ([17] page 10, Table 4).

The devices can operate on voltages between 2.6 V and 3.6 V, in our case the voltage is 3 V. ([16] page 2; [17] page 8, Table 2).

$I_{enc} = I_{dec} = i_{active} + i_{AES} =$ 15.7 mA + 223 µA. During coding and encoding the microcontroller and its AES module is working, because every packet is encrypted but there is no error correcting coding. The microcontroller operates on 32 MHz, with external clock on 3 V. ([16] page 63, Table 34-1).

$I_{tst} = I_{rst}$: 8.4 mA (CC1101) + 15.7 mA (XMega). In this state, the radio module runs frequency synthesizer (FSTXON state). The current draw equals in two cases: if the state changes from IDLE to RX or TX including calibration state. ([17] page 9, Table 4).

T_{tst}: 799 µs. The radio module needs time to switch to TX state including calibration. After transmission, it switches from TX to IDLE state and calibration takes negligibly little time (~0.1 µs). ([17] page 54, Table 34).

T_{rst}: 799 µs. The radio module needs time to switch to RX state including calibration. After transmission, it switches from RX to IDLE state and calibration takes negligibly little time (~0.1 µs). ([17] page 54, Table 34).

$T_{1enc} = T_{1dec}$: 1.465 µs/bit. The microcontroller performs AES coding in 16 byte units. For encoding or decoding a unit, 375 clock cycles are needed. Calculating with 32 MHz clock speed, this means 11.72 µs for 16 bytes, assuming data is bigger and neglecting padding of not exactly 16 bytes overhead, normalized for 1 bit it is 0.0916 µs/bit.

The power required by transmission, reception, encoding and decoding can be expressed as:

$$P_t = u\, i_{tx} = 3\ V \cdot 55.7\ mA = 167.1\ mW$$
$$P_r = u\, i_{rx} = 3\ V \cdot 35.7\ mA = 107.1\ mW$$
$$P_{tst} = P_{rst} = u\, i_{tst} = 3V \cdot 24.1\ mA = 72.3\ mW$$
$$P_{enc} = P_{dec} = u\, i_{enc} = 3\ V \cdot 15.92\ mA = 47.76\ mW$$

3.1. Forward Error Correction Schemes

The authors have chosen to use block codes for FEC, because their implementation uses fewer resources –from the limited computational capacity of microcontrollers – than other more advanced codes. The following three error correction codes were considered:

Hamming codes [6] are basic linear block codes [7] using parity checking as the added redundant information. They can only correct one bit per block and detect 2 incorrect bits. Hamming codes are perfect codes [7] and can be decoded using syndrome decoding [18]. They are often used in ECC memory modules.

Reed-Solomon [19], [20] codes are cyclic BCH codes. They are commonly used in CDs and DVDs.

BCH (Bose–Chaudhuri–Hocquenghem) [21] codes are also linear block codes, which can be defined by a generator polynomial.

To calculate the energy consumption of a FEC scheme, first the execution time of every FEC scheme on the same computer using Matlab simulation was measured. We chose this platform, as most of the FEC codes are already built-in. Then we implemented the selected code of each FEC scheme on the chosen microcontroller (Atmel AVR Xmega128 A3 [16]) and measured the clock cycles of executing encoding and decoding. Using our simulation data we could determine the proportion of each code and scaled the energy consumption according to the microcontroller's clock cycles.

The parameter κ_4 denotes the required energy for encoding and decoding in J/bit, and t is the error correcting capability in bit/block.

$$\kappa_4 = \frac{u\, i_{active}\, cycles_{FEC}}{N\, f_{CPU}} \left[\frac{J}{bit}\right] \quad (1)$$

During the selection process of the investigated schemes, among the many possible combinations of message lengths in a particular scheme, the best performer was chosen as our comparisons show on Fig. 2 - Fig. 4.

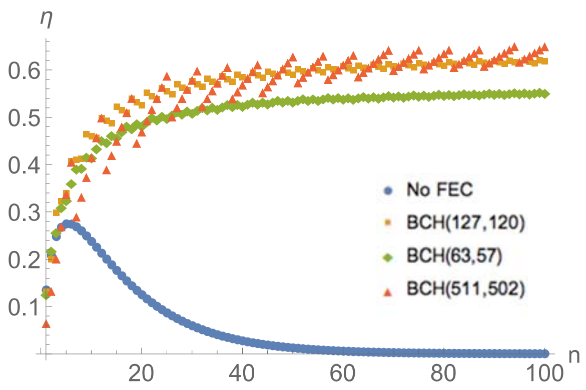

Fig. 2. Comparison of BCH codes.

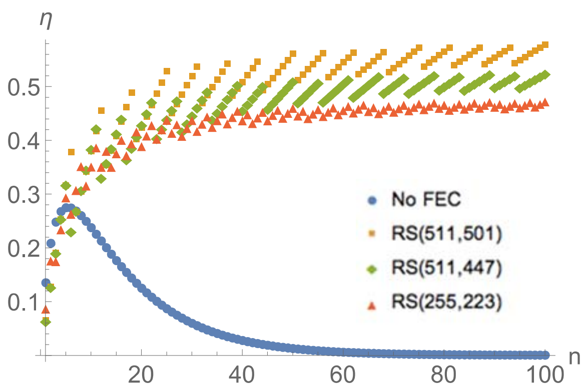

Fig. 3. Comparison of Reed-Solomon codes.

The best performers among each type of block codes were the following: BCH (511, 501), RS (511, 501) and Hamming (255, 247). These calculations were executed on a p=10^{-3} BER channel, which is a rather noisy channel and the use of FEC schemes can be significantly more efficient.

Table 2 shows the important parameters of the FEC codes, which are used in the following calculations.

Each error correcting code was implemented on the Atmel microcontroller and the exact clock cycle count for encoding and decoding was measured and summarized in Table 3.

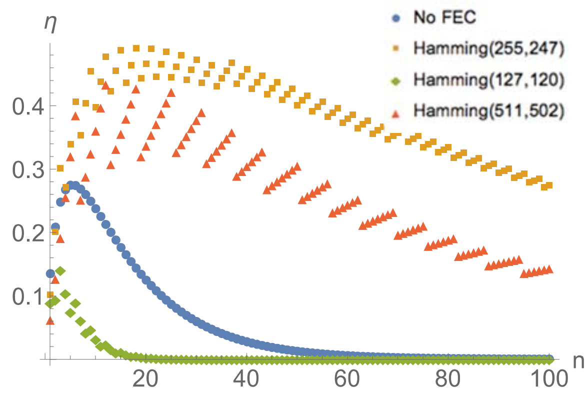

Fig. 4. Comparison of Hamming codes.

Table 2. Summary of FEC code parameters.

Code	Complex.	Type	N	K	t
No FEC	none	none	1	1	0
Hamming (255,247)	low	block	255	247	1
Reed-Solomon (511,501)	high	block	511	501	5
BCH (511,502)	high	block	511	502	4

Table 3. FEC codes clock cycle counts.

Operation	Clock cycle count	κ_4
RS encode	200000	5.0522 E-09
RS decode	510000	
BCH encode	130000	5.4344 E-07
BCH decode	970000	
Hamming encode	2000	1.7619 E-05
Hamming decode	28500	

3.2. Packet Error Rate

One way to describe the reliability of the radio channel is to calculate the Bit Error Rate (BER), which shows the amount of changed bits during transmission. In this scenario we communicate with packets and prefer to calculate whether a packet is corrupted in case of a certain BER, which can be expressed by the Packet Error Rate (PER).

In the calculation of PER we assume, that some kind of FEC is applied to correct statistically independent bits of the corrupted packet, and some kind of MAC is used to recognize malicious modifications of the payload. This paper does not take correlated bit errors into account. We also assume that FEC is not applied to the header of the packets so that no unnecessary calculations are made in case the destination address was corrupted. According to the previous assumptions the connection between the BER and the PER in case of FEC codes can be expressed as:

$$PER_{FEC} = 1-\left[(1-BER)^{\omega'}\left(\sum_{i=0}^{t}\binom{N}{i}BER^{i}(1-BER)^{N-i}\right)^{\left\lceil\frac{n\varphi+\omega_{MAC}}{K}\right\rceil}\right]. \quad (2)$$

In the calculation of the PER we used the ω' parameter for the overhead length, that includes the length of the ACK and assumed that the radio channel is symmetric for the BER.

Without the use of FEC (1) is simplified to (2), as the values of the parameters are: $N = 1$, $K = 1$ and $t = 0$ according to Table 2.

$$PER_{No\,FEC} = 1-(1-BER)^{\omega'+n\varphi+\omega_{MAC}} \quad (3)$$

4. Optimal Amount of Aggregation

To deal with the header and trailer of the packets together and to simplify the following equations let us introduce $\omega = \omega_h + \omega_{MAC}$ for expressing the overhead.

The amount of energy needed for sending and receiving one bit on a link without FEC can be calculated as:

$$E_b = \frac{E_t + E_r + E_{enc} + E_{dec}}{\omega + n\varphi} \quad (4)$$

The amount of energy needed for transmission is:

$$E_t = P_t\frac{\omega + n\varphi}{B} + P_{tst}T_{tst} \quad (5)$$

The amount of energy needed for reception is:

$$E_r = P_r\frac{\omega + n\varphi}{B} + P_{rst}T_{rst} \quad (6)$$

In this scenario the packets are sent encrypted by a built-in AES module, and Message Authentication Code (MAC) is employed to ensure integrity. Therefore the coding and decoding procedure consist of two phases: the MAC is calculated for the entire $\omega_h + n\varphi$ bit long packet, but only the $n\varphi$ bit long payload is encrypted to ensure that the headers are easily accessible for faster packet processing and routing. According to the previous lines the energy needed for encoding and decoding can be expressed as:

$$E_{enc} = P_{enc}(\omega_h + 2n\varphi)T_{1enc}, \quad (7)$$

$$E_{dec} = P_{dec}(\omega_h + 2n\varphi)T_{1dec}$$

Substituting (4)-(6) into (3) we get

$$E_b = \frac{P_t\frac{\omega + n\varphi}{B} + P_{tst}T_{tst}}{\omega + n\varphi} + \frac{P_r\frac{\omega + n\varphi}{B} + P_{rst}T_{rst}}{\omega + n\varphi} + \frac{P_{enc}(\omega_h + 2n\varphi)T_{1enc}}{\omega + n\varphi} + \frac{P_{dec}(\omega_h + 2n\varphi)T_{1dec}}{\omega + n\varphi} \quad (8)$$

Three new parameters are introduced to group the energy consumption parameters by functionality:

$$\kappa_1 = \frac{P_t + P_r}{B} = \frac{167.1\text{ mW} + 107.1\text{ mW}}{9600\,\frac{\text{bit}}{\text{s}}} = 28.56\,\frac{\text{mJ}}{\text{bit}},$$
$$\kappa_2 = P_{enc}T_{1enc} + P_{dec}T_{1dec} = 47.76\text{ mW}\cdot 0.0916\,\frac{\mu\text{s}}{\text{bit}} + 47.76\text{ mW}\cdot 0.0916\,\frac{\mu\text{s}}{\text{bit}} = 8.75\,\frac{\text{nJ}}{\text{bit}}, \quad (9)$$
$$\kappa_3 = P_{tst}T_{tst} + P_{rst}T_{rst} = 72.3\text{ mW}\cdot 799\ \mu\text{s} + 72.3\text{ mW}\cdot 799\ \mu\text{s} = 57.77\ \mu\text{J}$$

Using these parameters, the E_b energy required for sending and receiving one bit can be rephrased as:

$$E_b = \kappa_1 + \frac{\alpha_2\,\kappa_2 + \kappa_3}{\omega + n\varphi}, \quad (10)$$

where $\alpha_2 = \omega_h + 2n\varphi$.

Taking into account, that each packet needs an ACK, (which is ω_h bit long) to confirm successful delivery:

$$\kappa_3' = 2\kappa_3 = 115.54\ \mu\text{J}$$
$$\omega' = 2\omega_h \quad (11)$$

Assuming that the sent packets arrive successfully with probability $1 - PER$ on a channel characterized by certain PER, the probability of successful reception increases with the number of retransmissions. The probability, that the number of retransmissions until success will be k, is given by probability variable X with geometric distribution and $p = 1 - PER$

$$P(X = k) = PER^{k-1}(1 - PER) \quad (12)$$

The expected value of X – which means that in an average how many packets need to be sent for a successful reception – can be expressed as (according to geometric distribution):

$$E(X) = \sum_{k=1}^{\infty} k\cdot P(X = k) = \frac{1}{1 - PER} \quad (13)$$

Using (12), r which denotes the average number of required retransmissions (for the channel characterized by PER) can be determined. The value of r should be a positive integer ($r \in \mathbb{Z}^+$), because every fraction of packet sent is considered to be a part of a new packet, therefore:

$$r = \left\lceil \frac{1}{1 - PER} \right\rceil \tag{14}$$

According to (8), (10) and (13) the required energy for sending and receiving a packet is:

$$E_{req} = r\,(\alpha_1 \kappa_1 + \kappa_3') + \alpha_2\,\kappa_2 + \alpha_4 \kappa_4, \tag{15}$$

where $\alpha_1 = \omega + \omega_h + N\left\lceil \frac{n\varphi}{K} \right\rceil$, $\alpha_4 = \omega_{MAC} + n\varphi$.
Equation (13) can be grouped as:

- $\alpha_1 \kappa_1$ is the energy required for transmission and reception;
- κ_3' is responsible for switching RX and TX states;
- $\alpha_2 \kappa_2$ is used for encoding and decoding using AES and calculating MAC, and finally;
- $\alpha_4 \kappa_4$ is the amount of energy used for calculating FEC.

According to (13) the process of coding and decoding is executed once for every packet for the necessary number of bits (in case of MAC: the header and the payload; in case of encryption and decryption: only for the payload).

Besides, because of packet loss, all the packets and their ACKs should be sent *r* times in average to ensure ε probability of success (also switching RX-TX states should be done *r* times).

Remark. In this paper we ignored methods to counter replay attacks, because there are solutions, which change the number of bits present in the header, therefore our calculations should also depend on them.

Now having these formulas, we evaluate the usage of packet aggregation and FEC in parallel, and determine the amount of energy saved using them considering a certain BER of the radio channel.

Let $E^1_{No\,FEC}$ refer to the energy consumed during sending and receiving a packet without aggregation and FEC. Let us calculate the amount of gains we can achieve using aggregation and FEC compared to no aggregation and no FEC as a baseline

$$\Theta = \frac{n\,E^1_{No\,FEC}}{E^n_{FEC}}, \tag{16}$$

where E^n_{FEC} denotes the energy needed for sending an *n*-aggregated packet using FEC. The ratio expressed in (15) was determined for the discussed three FEC codes. The parameters of these FEC codes can be found in Table 2.

4.1. Results

We introduced the protocols and corresponding parameters in the previous sections. To demonstrate the consequences of the formulas and to determine the possible amount of energy that can be saved, the calculations are performed on the parameters of a real system developed by the authors. Among these parameters some characterize the hardware, while others describe the protocol.

Fig. 5 shows the gain Θ (the ratio of not using aggregation and using it) that can by achieved by using aggregation without FEC. The graph line representing $BER = 4 \cdot 10^{-3}$ is jagged, because the number of required retransmissions is growing as the aggregation number n is increasing. The number of retransmissions is the same in the neighbouring points, which follow each other without a jump in their values. The reason why the results achieved by using aggregation is better compared to the *n*-packet based algorithm is, that we lose the overhead of headers.

In the figures of this section, the $\Theta = 1$ marks the level above which the use of aggregation is more efficient.

Remark. This phenomena can be observed in case of other BER values, e.g. for $BER = 4 \cdot 10^{-5}$ *the first jump is at* $n \approx 200$, *which is above the aggregation value we considered worthy to examine.*

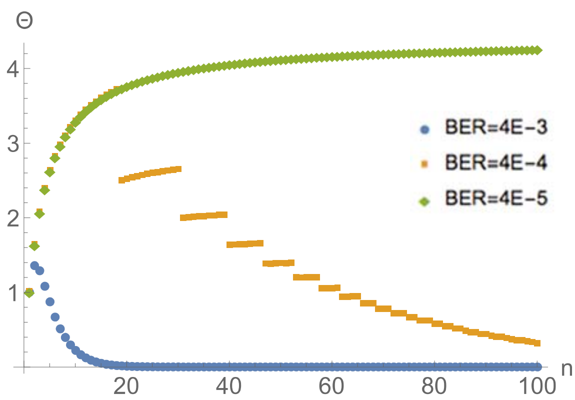

Fig. 5. Θ for different BER values without FEC.

Fig. 6 has the same setup as Fig. 5, with the only difference that Hamming codes were applied. The graphs show that in case of medium quality channel ($BER = 4 \cdot 10^{-4}$) and good quality ($BER = 4 \cdot 10^{-5}$) channel, there is no difference; the calculated points are perfectly aligned.

Fig. 7 and Fig. 8 show similarity of the values of Θ for different BER levels with respect to aggregation number n for Reed-Solomon and BCH FEC codes.

Analysing Fig. 7 and Fig. 8 it can be noticed, that in case of a poor quality channel ($BER = 4 \cdot 10^{-3}$) for every aggregation number *n* we got better gain Θ values then in case of better channel. This is because more powerful FEC codes provide more benefits compared to the same aggregation numbers in case of poor quality channels. Better BER channels achieved the same gain Θ.

Also, the graphs are looking like stages because the block length of Reed-Solomon codes is fixed. Therefore if the payload is not long enough padding is used to fill the rest of the block, which is inefficient.

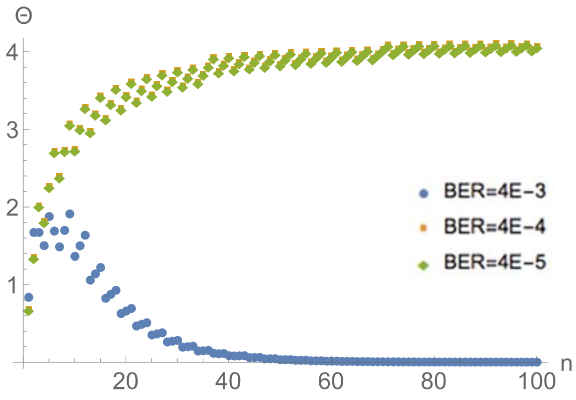

Fig. 6. ***Θ*** for different BER values with Hamming code.

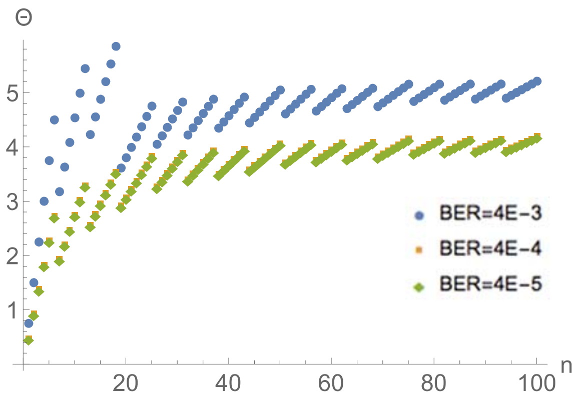

Fig. 7. *Θ* for different BER values with Reed-Solomon code.

In Fig. 8 in case of aggregation number $n < 40$ the poor quality channel gains more using aggregation and BCH code, than the better quality channels. Also at better quality channel there is significant gain compared to baseline (no aggregation, no FEC) just like in the case of Reed-Solomon codes.

The next three figures (Fig. 9 - Fig. 11) compare the cases of different FEC codes grouped by channel quality ($BER = 4 \cdot 10^{-3}, 4 \cdot 10^{-4}$ and $4 \cdot 10^{-5}$) in respect to n. For every diagram, a table is included, which shows the optimal aggregation number (the highest point of the graphs and the corresponding number of required retransmissions.

Remark. The optimal aggregation number can be much higher is case of BCH and RS codes, but the authors considered n<100 aggregation numbers are worth dealing with, because higher aggregation numbers would cause much higher delays. For example if the aggregation number n=100 and the packets are generated on an hourly base, then the aggregation delay can be as high as 100 hours. For most real-world scenarios the delay should be within a day.

Fig. 9 compares FEC codes on the worst quality channel. This scenario shows the energy cost of different FEC codes the best. The graph emphasizes, that not using any FEC is the worst, and BCH and Reed-Solomon codes perform as the best. It can be seen, that in case of lower aggregation numbers ($n < 10$), Reed-Solomon is the best solution, and from $20 < n < 40$ RS and BCH are at the same level. When further increasing the aggregation number RS code is the most efficient again.

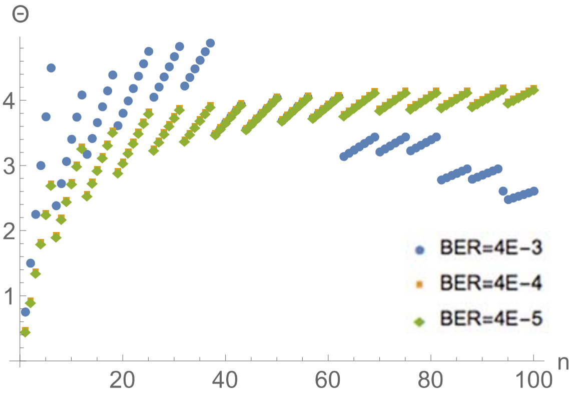

Fig. 8. *Θ* for different BER values with BCH code.

	No FEC	Hamming	BCH	RS
Opt. aggr. no.	2	9	37	18
No. of reps.	6	8	4	3

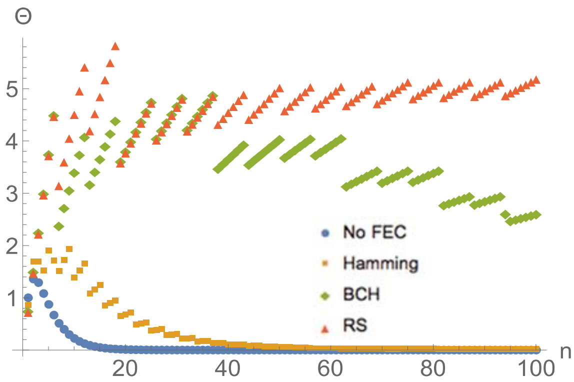

Fig. 9. Comparison of FEC codes at BER=4E-3.

Fig. 10 compares FEC codes on a channel with $BER = 4 \cdot 10^{-4}$. It can be seen, that in case of *n>20* aggregation numbers, FEC codes provide more energy efficient operation. The FEC codes perform roughly the same.

	No FEC	Hamming	BCH	RS
Opt. aggr. no.	18	98	94	100
No. of reps.	2	2	2	2

According to Fig. 11 in good quality channels there is no benefit of using FEC codes, because for every aggregation number the case without FEC performs as the best. The FEC codes just converge to the graph of no FEC case.

	No FEC	Hamming	BCH	RS
Opt. aggr. no.	100	98	94	100
No. of reps.	2	2	2	2

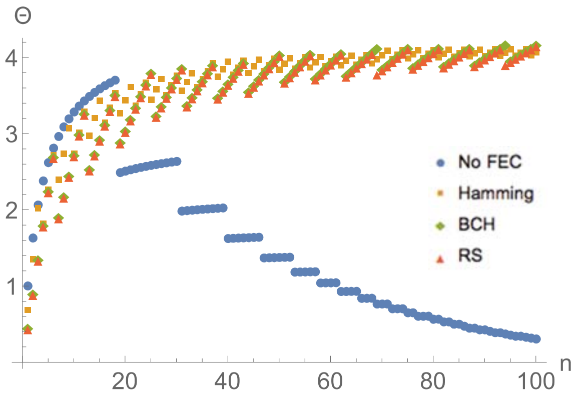

Fig. 10. Comparison of FEC codes at BER=4E-4.

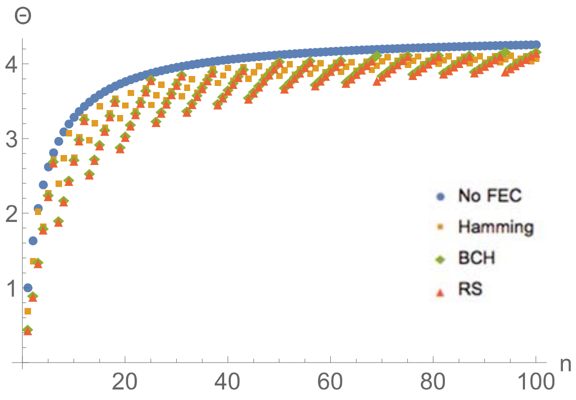

Fig. 11. Comparison of FEC codes at BER=4E-5.

Fig. 12 shows the values of Θ as function of n and BER. It can be seen that Reed-Solomon codes result in the best performance for this n and BER range.

5. Conclusions

This article discussed the energy efficiency of different Forward Error Correction algorithms and presented a method to calculate the optimal amount of aggregation of the data packets in terms of power consumption, while taking into account the Bit Error Rate characteristics of the wireless channel. With the help of the methods shown in this paper, developers and researchers can optimize the energy consumption of their wireless sensor network protocol.

Acknowledgements

This research has been supported by BME-Infokom Innovator Nonprofit Ltd. http://www.bme-infokom.hu

This research has been sponsored by The European Union's Hungary-Slovakia Cross-border Co-operation Programme. Building Partnership. www.husk-cbc.eu, www.hungary-slovakia-cbc.eu The content of this paper does not necessarily represent the official position of the European Union.

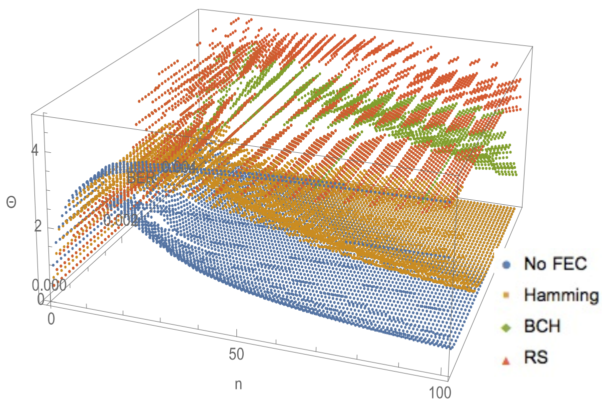

Fig. 12. Θ as function of n and BER.

References

[1]. Á. Milánkovich, G. Ill, K. Lendvai, S. Imre, S. Szabó, Research on Payload Aggregation of Packets in WSNs, in *Proceedings of the 4th International Conference on Sensor Networks,* Angers, February 2015, pp. 187-194.

[2]. Y. Sankarasubramaniam, I. F. Akyildiz, S. W. McLaughlin, Energy efficiency based packet size optimization in wireless sensor networks, in *Proceedings of the First IEEE International Workshop on Sensor Network Protocols Applications*, 2003, pp. 1–8.

[3]. Z. Tian, D. Yuan, Q. Liang, Energy Efficiency Analysis of Error Control Schemes in Wireless Sensor Networks, in *Proceedings of the IEEE International Wireless Communications and Mobile Computing Conference,* Crete, Greece, 2008, pp. 401–405.

[4]. P. Kumar, M. Jayakumar, Comparison of Bit Error Rate for Propagation Mechanisms of Millimeter Waves in a Practical Communication Systems Employing PSK FSK, in *Proceedings of the Progress in Electromagnetics Research Symposium (PIERS)*, Jul. 2010, pp. 292–295.

[5]. G. Balakrishnan, M. Yang, Y. Jiang, Y. Kim, Performance Analysis of Error Control Codes for Wireless Sensor Networks, in *Proceedings of the 4th IEEE International Conference on Information Technology (ITNG '07),* 2007, pp. 876–879.

[6]. S. Lin, D. J. Costello, Error Control Coding: Fundamentals and Applications, *Prentice Hall*, 2004.

[7]. T. Etzion, A. Vardy, Perfect binary codes: constructions, properties, enumeration, *IEEE Transactions on Information Theory*, Vol. 40, No. 3, May 1994, pp. 754–763.

[8]. D. Wei, H. A. Chan, Clustering Ad Hoc Networks: Schemes Classifications, in *Proceedings of the 3rd Annual IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON '06),* Vol. 3, 2006, pp. 920–926.

[9]. M. C. Vuran, I. F. Akyildiz, Cross-Layer Packet Size Optimization for Wireless Terrestrial, Underwater, Underground Sensor Networks, in *Proceedings of the*

IEEE Conference on Computer Communications (INFOCOM'08), Jan. 2008, pp. 1–9.

[10]. M. C. Vuran, I. F. Akyildiz, Cross-Layer Analysis of Error Control in Wireless Sensor Networks, Vol. 2. in *Proceedings of the 3rd Annual IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON '06),* 2006, pp. 585–594.

[11]. L. Galluccio S. Palazzo, End-to-End Delay Network Lifetime Analysis in a Wireless Sensor Network Performing Data Aggregation, in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'09)*, Apr. 2009, pp. 1–6.

[12]. H.-H. Yen, Optimization-based channel constrained data aggregation routing algorithms in multi-radio wireless sensor networks, *Sensors*, Vol. 9, No. 6, Dec. 2008, pp. 4766–4788.

[13]. J. Geibig, D. Bradler, Self-organized aggregation in irregular wireless networks, *Wireless Days (WD), IFIP'2010,* 2010, pp. 1–7.

[14]. K. Lendvai, Á. Milánkovich, S. Imre, S. Szabó, Optimized packet size for energy efficient delay-tolerant sensor networks, in *Proceedings of the 8th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* Barcelona, 2012, pp. 19–25.

[15]. Optimized Packet Size for Energy Efficient Delay-Tolerant Sensor Networks with FEC, in *Proceedings of the 12th International Conference on Telecommunications (ConTEL),* Zagreb, 2013, pp. 87–94.

[16]. Atmel, 8/16-bit XMEGA A3, Microcontroller, June 2013, pp. 1–134.

[17]. CC1101 Low-Power Sub-1 GHz RF Transceiver (Rev. I), *Texas Instruments, Inc.*, Nov. 2014.

[18]. M. P. C. Fossorier, S. Lin, J. Snyders, Reliability-based syndrome decoding of linear block codes, *IEEE Trans. Inform. Theory*, Vol. 44, No. 1, January 1998, pp. 388–398.

[19]. D. V. Sarwate N. R. Shanbhag, High-speed architectures for Reed-Solomon decoders, *IEEE Transactions on VLSI Systems*, Vol. 9, No. 5, October 2001, pp. 641–655.

[20]. S. B. Wicker, V. K. Bhargava, Reed-Solomon codes their applications, *IEEE Press,* 1999.

[21]. R. C. Bose, D. K. Ray-Chaudhuri, Further results on error correcting binary group codes, *Information and Control*, Vol. 3, No. 3, Dec. 1959, pp. 279–290.