SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 10, No. 3, October 2013, 399-414

UDK: 654.16:004.78]:681.586 **DOI: 10.2298/SJEE130924012K**

Using Duty Cycle Extension in Time Synchronization and Power Saving

Mirko Kosanović[1], Mile Stojčev[2]

Abstract: Sensor Nodes, as constituents of Wireless Sensor Network, are battery-powered devices and have limited amount of energy available. This limitation must be overcome by using techniques that reduce the consumption of energy effectively. In this article we present a technique called duty cycle extension. It relates to modification of a guard time, and is intended to compensate SN's quartz frequency instability. In a case when the packet rejection ratio is large, from 40% up to 70%, our intent is to show how duty cycle extension affects both time synchronization, and SN's battery energy budget. We study how time scales affect duty cycle ratio by varying the time interval between packet transmissions over three orders of magnitude, from 1 s to 3600 s. By implementing this method, an increase of battery capacity from 0.01 % up to 1.406 %, per single year working period, is needed.

Keywords: Duty cycling, Energy efficiency, Lost packets, Time synchronization, Wireless sensor networks.

1 Introduction

Wireless Sensor Networks (WSNs) represent a set of tiny, inexpensive sensor nodes (SNs). The main source of power supply of SN is usually battery, which has a strictly limited capacity. Therefore, the need for efficient power management becomes imperative. In general, there are two solutions of this problem. The first relates to finding effective methods for optimal use of available energy, while the second one deals with the usage of natural resources as additional sources of energy, i.e. implementation of energy harvesting techniques [1].

In this paper we focus on implementation of a technique called duty cycle extension as an approach intended to achieve efficient usage of available energy. Implementation of duty cycling means that a SN most of the time aggressively switches off its electronics and does essentially becomes disconnected from the network. In this dynamic environment, it is a major challenge to provide correct connectivity to the time-synchronized WSN, besides minimizing the energy consumption. As a consequence of a SN's local

[1]Technical College of Professional Studies Niš, Serbia; E-mail: mirko.kosanovic@open.telekom.rs
[2]University of Niš, Faculty of Electronic Engineering Niš, Serbia

oscillator frequency instability, which causes an error in time synchronization, we introduce the duty cycle extension in order to compensate this error. Even in a case when both there is a large percentage of incorrectly received packets as well as several consecutive packets are lost, by using the proposed method we provide (enough wide) time–slot within which we expect to receive the packet. The cost which we pay for involving duty cycle extension is really a minor increase of battery capacity (< 1.406% per year).

2 WSN and Sensor Node Characteristics

Before we start with explanations how to cope with the problem of saving energy, it is necessary first to point out to some details related to: i) identifying basic building blocks of the SN architecture; and ii) specifics of WSN operation that have dominant impact on power consumption.

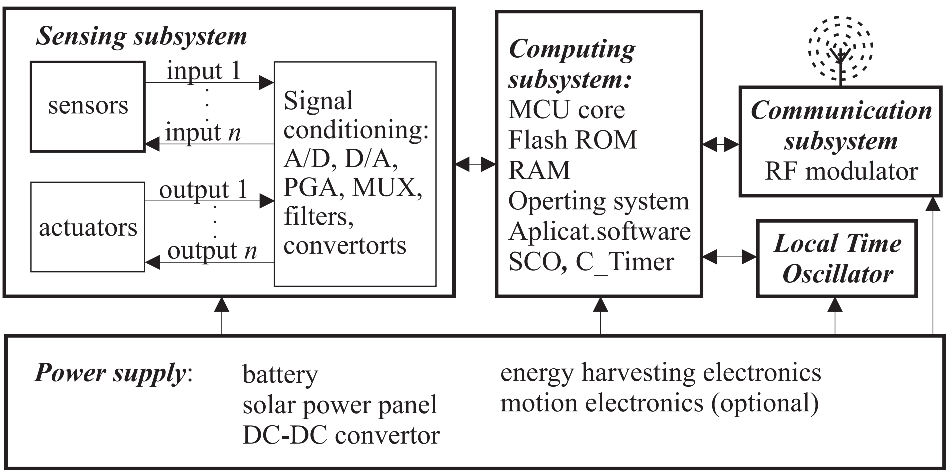

Fig. 1 – *System architecture of a typical sensor node.*

2.1 Sensor node architecture

The SN consists of several building blocks: power supply, sensing, computing, communication, and optional: mobile unit, coordinate unit, time system synchronizer, etc. (Fig. 1). Due to limited amount of battery capacity, SNs are characterized by a restricted number of hardware resources (sensors, memory capacity, input-output peripherals, etc.). From power consumption point of view, the communication task, in respect to sensing and data processing, is a more demanding one [2]. More details about SN architectures and functionalities of their building blocks can be found in [3].

2.2 Inaccuracy in timestamp generation and synchronization

Time record within SN architecture (see Fig. 1.) is provided by clock oscillators and timers. Two different clock oscillators, referred as system clock oscillator (SCO) and local time oscillator (LTO), typically drive the SN. The LTO operates without interruption. The reason for using two separate clock generators within SN is the following: The LTO operates continually in time, contrary to SCO, which decreases or stops CPU activities when enters in different power saving modes. During power saving mode the CPU loses all timing information. Therefore, the SCO for time keeping by itself cannot be used [4].

Most hardware oscillators have not so stable and precise frequency. This means that they never generate accurate time intervals by which timestamps are created. Crystal oscillator ticks are of slightly different rate due to impact of manufacture techniques, ambient temperature, pressure, battery voltage, oscillator aging, and other effects. Let the crystal oscillator inaccuracy, s_X ($\Delta f/f$), be in the range from 1 ppm (parts per million) up to 50 ppm. This implies that if the nominal frequency of a crystal oscillator is only 1 MHz, the time between two SNs may drift (1-50) μs per second. As a direct consequence of s_X we have that very small frequency deviation would bring a time uncertainty of about 0.864 – 4.32 s per day [5].

Synchronized network time is an essential aspect for energy efficient scheduling and power management of SNs. It allows SNs to shutdown their RF transceivers and other peripherals, even microcontrollers, to enter into power saving mode, and later to return to normal operating mode. The following three solutions for time synchronization in WSN are used:

a) one-way packet dissemination;
b) receiver-receiver synchronization; and
c) two way packet exchange [6].

In this paper we use one-way packet dissemination because it is the simpler one.

2.3 Packet structure

The structure of a packet used in the proposed protocol is given in Fig. 2. Its structure is supported by large number of RF transceivers [7, 8], and consits of the following fields:

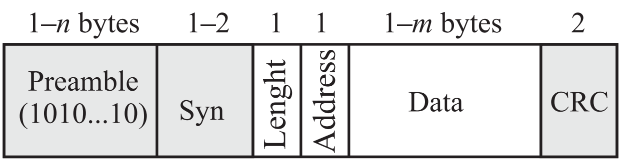

Fig. 2 – *Packet format.*

Preamble – Alternating sequence of ones and zeros (10101010...). The lenght of preamble field is programmable in multiple number of bytes. When the transmitter is switched-on, the RF modulator starts with sending preamble bytes.

Syn – Programmed number of synchronization bytes.

Lengh –t Specifies the lenght in bytes of payload data. This field is optional. If payload data is of fixed lenght this field is ommited.

Address – Specifies unique SN's address. This field is optional if address filtering is not used.

Data – Carries informations, including command and useful data. The lenght of this field is variable and is deffined by the field *Length*.

CRC – Two bytes field, used for detecting error during transmission. This field is optional.

2.4 Packet delivery

A comprehensive study which relates to empirical measurements of packet delivery performance with or without high power external noise in the network, are presented in [9]. Temporal and spatial correlations of packet delivery and link asymmetry, as three crucial characteristics of the network, are identified. Link definition is involved. A link is dead if it has a packet reception ratio (PRR) of 0 %, poor if the PRR is < 10%, intermediate when PRR is between 10% and 90%, good if PRR is between 90% and 100%, and perfect when PRR is 100% [9].

In this article, we will involve a factor called packet rejection ratio, PRJR, defined as PRJR [%] = 100 [%] – PRR [%]. It corresponds to the percentage of rejected packets. We consider a case when the WSN operates in environmental conditions such that the PRJR is between 70% and 40% [10]. Such WSN we find when high power external noise sources can cause packet delivery corruption, i.e. WSN operation is vulnerable to interference from cohabitating 802.11 and Bluetooth systems, radiation from high power microwave ovens, and other sources.

In our design solution, the packet is rejected when the wake-up time of the receiver occurs within the last preamble byte. In such scenario a correct in-time synchronization of the SCO cannot be achieved.

3 Duty Cycling

In WSN energy conservation is very important. The four main ways in which SN consumes energy are sensing, communication, computation, and storage [11]. For each unit of useful work it performs, the SN consumes different amount of energy. In order to minimize the energy consumption of SN, different techniques for reducing power consumption are used [2, 12]. Duty cycling is an effective and commonly used technique to lower the rate of energy consumption. The idea behind this is clear. Keep hardware in a low power sleep state except on infrequent instances when the hardware is needed. However, duty cycling leads to more complex communication patterns that include polling, and scheduling the channel. Radio operation dominates the SN power budget. In [2] it is shown that even at low duty cycle of approximately 1 to 2 %, radio operation is about one order of magnitude more expensive than all other operations combined. Hence, SN's lifetime improvements will require substantial reduction of radio on-time.

3.1 The functions of LTO and C_Timer

In many design solutions, duty cycling allows the processor to be put into a low power state for extended period. In [13] it is shown that during this period, only the LTO tracks time with aim to trigger the wake-up sequence. When duty cycle is low (< 1%), a high accurate and high frequency LTO is needed [14]. Such LTO can quickly become the most significant energy consumer during sleep time, thus invalidating any power gain made in synchronization accuracy. In order to cope with clock drift, less stable LTO requires SNs to synchronize more frequently. In principle, more stable clock can be used to improve duty cycling capabilities.

In our proposal, the LTO is used for determining duty cycle periods and creation of local timestamp, only. When the CPU enters into active mode all time delays are measured with resolution defined by the local SCO, i.e. the timer within the Computing subsystem, called C_Timer (see Fig. 1). In this way, we improve time resolution measurement and indirectly decrease the time synchronization error.

3.2 Duty cycle definition

It comes as no surprise that most research aiming at prolonging the lifetime of WSNs focuses on limiting the operation of some SN's constituents (radio, sensors, CPU, …). This implies intermittently switching the SN on and off. The periods during which the SN is on or off are known as its active (T_{ON}) and inactive (sleep) period (T_{OFF}), respectively. The complete time synchronization period, T_{Σ}, (see Fig. 3) is equal to:

$$T_{\Sigma} = T_{OFF} + T_{ON} . \tag{1}$$

The fraction of the time during which a SN is on, in respect to the complete time synchronization period, is referred to as duty cycle (DC), and is defined as:

$$DC = \frac{T_{ON}}{T_{\Sigma}} = \frac{T_{proc} + T_{guard}}{T_{proc} + T_{guard} + T_{OFF}} , \tag{2}$$

where T_{proc} corresponds to the processing time needed for sensing, data processing, data storing, and transmitting packets. T_{guard} represents a guard time.

During operation, the frequency skew results in relative clock drift between nodes. As a result, nodes must include a guard time. The guard time is equal to the maximum drift, and linearly depends on $T_\sum$. If s_X ($\Delta f/f$) is the frequency skew, and $T_\sum$ is the total time synchronization period, then the minimum guard time becomes $T_{guard} = 2T_\sum s_X$. SNs whose active periods do not overlap cannot communicate with each other.

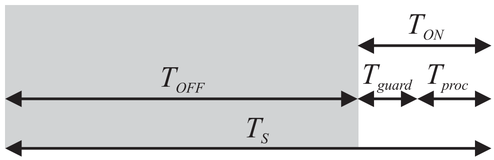

Fig 3 – *The complete time synchronization period.*

4 Performance Evaluation

In order to increase its lifetime, the SN should operate with minimum possible energy consumption. This requires power aware computation, low-energy signaling and networking, and power aware software communication. Lifetime refers to the period for which a SN is capable of sensing and transmitting the sensed data to the base station(s). The following two approaches are used for reducing energy consumption in SN:

i) *duty cycling* – consists of waking-up the SN for the time period needed to acquire a new set of samples, and then powering it off immediately afterwards, and

ii) *adaptive-sensing strategy* – is able to change dynamically the SN's activity to the real dynamics of the process. Our solution uses duty cycling because its implementation is simpler in respect to adaptive-sensing strategy.

In the sequel a detailed analysis which relates to determining the battery capacity of a SN, when the SN operates under harsh environment conditions with high percentage of lost packets, will be given.

4.1 Average current

In the proposed design the CPU of a SN is based on a low power microcontroller MSP430F123 and its communication part on the RF modulator CC 2420. The MSP430 runs in two operating modes, Active with 300 μA at 1 MHz and 3 V power supply, and Low Power Mode 3 with power consumption of 0.7 μA. MSP430 uses two quartz oscillators, LTO (f_{LTO} = 32 kHz) and SCO (f_{SCO} = 1 MHz). LTO is always active, while SCO is active during T_{proc} and T_{guard} periods. In transmitting mode the RF modulator consumes 17.4 mA, in receiving mode 19.7 mA, and in sleep mode 1 μA. Data transfer rate is 128 kbps and packet length is 64 byte. The initial duty cycle value, which is used as parameter for evaluating the power consumption, can be within a range from 1% down to 0.01%. The inaccuracy of LTO, s_{LTO}, in the worst case, is 50 ppm. The capacity of a Lithium-ion battery is 560 mAh. A sensor MS55ER for barometric-pressure, with average current consumption of 1 mA, is connected to the SN.

4.2 Ideal communications

Our analysis will start by considering ideal communications for which we adopt that there are not lost packets during data transfer, and T_{proc} is the same for all *DC* factors.

For T_{Σ} (see Fig. 3) we have:

$$T_{\Sigma} = T_{OFF} + T_{proc} + T_{guard} = T_{OFF} + T_{proc} + 2T_{\Sigma}s_x \,, \tag{3}$$

while

$$T_{OFF} = T_{\Sigma}(1 - 2s_x) - T_{proc} \,. \tag{4}$$

Time durations of T_{proc}, T_{guard}, T_{OFF}, and T_{Σ}, for different *DC* values and LTO oscillator inaccuracy $s_{LTO} = 50$ ppm, are given in **Table 1**. Note that, the unit increment for all time periods is determinined by the LTO (for 32 kHz, this increment corresponds to 30.5 μs).

Table 1

Time periods T_{guard}, T_{OFF}, and T_{Σ} for different DC factors and fixed T_{proc}.

DC [%]	T_{proc} [ms]	T_{guard} [ms]	T_{OFF} [ms]	T_{Σ} [ms]
0.05	10.004	2.5315	24997.4645	25010.0305
0.1	10.004	1.1285	11104.42306	11115.5725
0.5	10.004	0.2135	2031.415153	2041.6395
1	10.004	0.122	1000.379051	1010.526
5	10.004	0.0305	190.4464619	200.507
10	10.004	0.0305	90.10564014	100.162
50	10.004	0.0305	9.9775024	20.0385
100	10.004	0	0	10.004

As candidate for low power design, a system with lower *DC* factor is better choice. In this case, the SN more of its time interval T_{Σ} spends into sleep mode.

The average current I_{AVR}, during T_{Σ} is equal to:

$$I_{AVR} = \frac{T_{ON}(I_{ACPU} + I_{ARF} + I_{sen}) + T_{OFF}(I_{SCPU} + I_{SRF})}{T_{\Sigma}} \,, \tag{5}$$

where I_{ACPU}, I_{ARF}, I_{sen} and I_{SCPU}, I_{SRF} correspond to a current of the CPU, transceiver, and sensor, during T_{ON} and T_{OFF}, respectively. In our design the sensor electronics is active during T_{ON}, only. By substituting (2) into (5), we obtain:

$$I_{AVR} = DC(I_{ACPU} + I_{ARF} + I_{sen}) + (1 - DC)(I_{SCPU} + I_{SRF}) \,. \tag{6}$$

Table 2

Average current for different $T_{\sum}$ in trems of S_{LTO}.

$T_{\sum}$ [s]	I_{AVR} [μA]				
	10 ppm	20 ppm	30 ppm	40 ppm	50 ppm
1	212.407441	212.407441	212.407441	212.407441	212.407441
10	22.770744	22.770744	22.770744	22.770744	22.770744
30	8.723581	8.723581	8.723581	8.723581	8.723581
60	5.211791	5.211791	5.211791	5.211791	5.211791
600	2.051179	2.051179	2.052246	2.052246	2.052246
1800	1.817415	1.817771	1.818127	1.818483	1.818839
3600	1.758886	1.759241	1.759775	1.760131	1.760487

In **Table 2** the needed average currents, I_{AVR}, in terms of S_{LTO}, for different values of $T_{\sum}$ as parameters, are given. As we can seen from **Table 2** the impact of S_{LTO} to I_{AVR} for fixed $T_{\sum}$ is minor, i.e. less then 0.01%. For $S_{LTO} = 10$ ppm, the ratio between I_{AVR} for $T_{\sum} = 1$ s and I_{AVR} for $T_{\sum} = 3600$ s, denoted as $R_{AVR} = 212$ μA/1.7 μA ≈ 124.

4.3 Non ideal communications

The strength of electromagnetic signals (radiated power) decreases rapidly as the distance between the transmitter and receiver increases (signal attenuation is high compared to wired communication). The situation becomes worse when the sender operates in error prone environments and for links operating close to noise floor. In this case, the receiver may receive signals not only from the intended transmitter, but also from other transmitters if they are using the same carrier frequency, i.e. interference appears due to shareable channels of coexisting 802.11b or Bluetooth networks. For example, packet losses are because 802.11b nodes usually do not defer transmission when a SN packet transmission is in progress. In [9] it is presented that 802.11b transmission power is larger than that of SN by a factor of 100. Strong electromagnetic disturbances in industrial plants often may come from radiation of high power microwave ovens, too. In addition, the receiver may receive more than one signal from the same transmitter because electromagnetic waves can be reflected back from obstacles such as walls, the ground, or other objects. The result is that a receiver accepts several signals with different phases, i.e. multipath propagation exists. This makes the signal less recognizable.

Packet delivery is a result of the signal to noise ratio (SNR). The SNR describes how strong the intended RF signal is in comparison to the additive Gaussian noise. SNs with build-in RF modulator CC2420 usually use the receive signal strength indicator (RSSI) as the estimator of the link quality. In general, links with an average RSSI above –87 dBm are good links. Below this threshold, there is no clear correlation to RSSI.

During our previous analysis, we assumed that the communication is ideal and that, during data transfer, errors do not appear. In [10] it is presented that in industrial environments the PRJR is in the range from 40% up to 70%. Therefore, some modifications of time synchronization in the receiving process are necessary. In principle, there are two solutions for this problem. In the first one, when the SN does not receive a synchronization packet in time, it updates its local time according to the value obtained during the last correctly received packet. This solution is simple but has one serious drawback. Namely, the T_{ON} can slide out of borders, within which it is expected to appear, when several consecutive packets are not correctly received.

We use the second approach. In this case, when the SN does not receive packet in time, it updates its local time according to the value obtained during the last correctly received packet. In addition, in the next synchronization cycle the guard period T_{guard} is extended for $T_{ext} = T_{err} + T_F$, where T_F is a time delay involved by transmitter. It includes a time delay starting from reading timestamp until loading it into RF buffer. Let note that T_{guard} is used to compensate s_{LTO}, while by involving T_{ext} we can compensate now both a time delay needed for creating the timestamp during T_{proc} and T_{err} (as a consequence of s_{SCO}). The duration of T_F is several clock cycles (in our case it is 53 CPU clock cycles). For $f_{SCO} = 1$ MHz, and 64 byte long packet, we have $T_F = 0.053$ ms. For $T_{proc} = 20$ ms (a needed time to receive the timestamp, the updated time and to send a packet), $f_{SCO} = 1$ MHz, and $s_{SCO} = 50$ ppm, we obtain $T_{err} = 1$ µs.

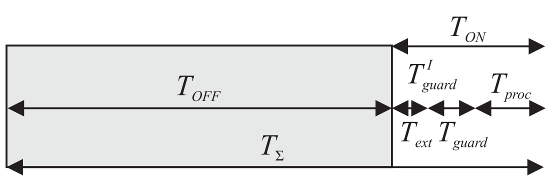

Fig. 4 – *The time synchronization period with T_{ext}.*

By involving a time correction T_{ext} ($T_{ext} = 0.054$ ms) within T^{I}_{guard} (see Fig. 4) we provide a time slot extension for T_{ON}, what enable us to decrease the impact of consecutively lost packets to a time synchronization process (for more details see Subsection 4.4).

During our analysis, we assume that the probability density function (PDF) of all missed packets corresponds to Binomial one, because it is more often used as a PDF function for lost packets in wireless communication, while for consecutively missed packets we use Normal Gaussian distribution, since it is used as standard distribution of the received packets in wireless communications [15, 16].

According to the aforementioned, (3) can be written now as:

$$T_{\Sigma} = T_{OFF} + T_{proc} + T^{I}_{guard} = T_{OFF} + T_{proc} + 2T_{\Sigma}s_x + \frac{\sum_{i=1}^{n} iv_i k(T_{err} + T_F)}{m}, \quad (7)$$

where:

- the limit value n depends on the value of a standard deviation, σ;
- $T^{I}_{guard} = T_{guard} + T_{ext}$, T_{ext} corresponds to the average error during receiving all packets;
- i is the number of consecutively missed packets;
- v_i is the probability of consecutively missed packets;
- k is a total number of lost packets; and
- m corresponds to the total number of delivered packets.

4.4 Total versus lost packets

Let consider a data transfer efficiency of the SN in harsh industrial working conditions (40% < *PRJR* < 70%). As a direct consequence of unreliable data transfer, large amount of limited battery energy is irretrievably lost. Therefore, analyze now in which way the number of lost packets have impact to the SN's lifetime. To this end, it is necessary to determine the needed battery capacity, BC, for single year SN's working period. The conducted analysis is general enough and can be applied to a period of several years, too. Our analysis takes into account the effect of self-discharging current, I_{SD}, too. For instance, for chargeable (non-chargeable) lithium-ion battery, I_{SD} causes BC loss of 10% (2%) per year [17]. We will consider the worst case what corresponds to BC lost of 10% per year. During this, all working parameters (I_{AVR}, T_{on}, T_{off}, power dissipation of a CPU, RF module and sensor) retain the same values as those defined in Subsection 4.1. Firstly, by using Binomial distribution we have determined the maximal number of lost packets for four PRJR values. This analysis was performed for inter-packet-intervals (IPIs) of 1 s, 60 s and 3600 s (i.e. the packets were delivered every second, every minute and every hour).

Values that relate to the maximal number of lost packets for different *PRJR*s ($PRJR_i$, $I = 1, 2, 3, 4$) as parameter, and for different IPIs, $T_\Sigma = 3600$ s, $T_\Sigma = 60$ s, and $T_\Sigma = 1$ s, respectively, are given in **Table 3**.

Table 3

Total number of transmitted packets and the maximal number of lost packets for different PRJRs during one year period.

Period T_Σ	*Total number of packets*	*Lost packets*
$PRJR_1 = 70\%$		
$T_\Sigma = 1$ h	8760	6460
$T_\Sigma =60$ s	525600	370487
$T_\Sigma = 1$ s	31536000	22095108
$PRJR_2 = 60\%$		
$T_\Sigma = 1$ h	8760	5609
$T_\Sigma =60$ s	525600	318107
$T_\Sigma = 1$ s	31536000	18942885
$PRJR_3 = 50\%$		
$T_\Sigma = 1$ h	8760	4742
$T_\Sigma =60$ s	525600	265605
$T_\Sigma = 1$ s	31536000	15789726
$PRJR_4 = 40\%$		
$T_\Sigma = 1$ h	8760	3861
$T_\Sigma =60$ s	525600	212990
$T_\Sigma = 1$ s	31536000	12635689

4.5 Distribution of consecutively lost packets

The amount of a *DC* factor has direct impact on the SN's power consumption. T^I_{guard} (see Fig. 4), is component of variable time duration and in our proposal, according to (7), directly depends on to the number of consecutively lost packets. As we have already mentioned, in order to determine the probability of consecutively lost packets we use Normal Gaussian distribution. The results which relate to the number of consecutively lost packets for different values of standard deviation, σ, are presented in Fig. 5. σ corresponds to the number of packets that deviate from average value in respect to the total number of lost packets.

By analyzing the results presented in Fig. 5 we can conclude that: each lost packet have impact on error appearance in time synchronization. This error increases as the number of consecutively lost packets increases.

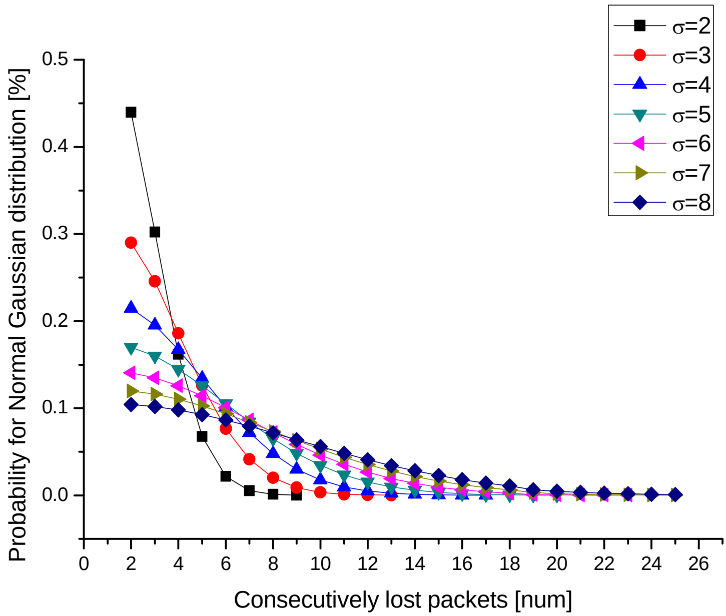

Fig. 5 – *Probability of consecutively lost packets for Normal Gaussian distribution for different σ as parameter.*

Probabilities of consecutively lost packets for Normal Gaussian distribution for different σ are sketched in Fig. 5, too. In general, curves denoted with lower σ values (see Fig. 5), are typical for shorter distance between the sender and receiver. This means that for shorter distances the dispersion of consecutively lost packets is smaller (i.e. narrower). Without loss of generality, we take that the maximal number of consecutively lost packets is $n_{max} = 26$, because the probabilities for all σ values are almost equal to zero for $n > 26$ (see Fig. 5). Therefore, in the current software version of our protocol the implemented algoritm declares some SN as disconnected after $n = 26$ consecutively lost packets.

4.6 Power consumption and battery capacity

Taking into account the results presented in **Table 3** and Fig. 5, according to (6) and (7), we can determine now in which way the number of lost packets have impact on SN's power consumption. Our analyze includes SN's power consumption with $f_{SCO} = 1$ MHz, for working period of one year, with different $T_{\sum}$ (1 s, 1 min, 1 h), and for standard deviations from $\sigma = 2$ up to $\sigma = 8$. The obtained results which relate to the increment of battery capacity (*BC*), when T_{ext} is included, are given in **Table 4**. Our analysis takes now into account the consumptions of SN during communication, sensing and data processing.

According to Fig. 2, correct time synchronization can be achieved if the receiving SN accepts at minimum one preamble byte per packet. If we assume that the transmission speed is 128 kbps, $s_{SCO} = 50$ ppm, the preamble is one byte length, and that a duty cycle extension is not included ($T_{ext} = 0$), then after two consecutively lost packets the time synchronization between two neighboring SNs will fall-out, i.e correct packet delivery will be not possible (The time period of s_{LTO} is 30.5 µs. For sending 8 bits at data rate at 128 kbps, 61 µs is needed. Since in our solution $T_{ext} = 53$ µs, then 53 µs + 30.5 µs > 61 µs, what implies that when more than one packet is lost, asynchronous communications instead of synchronous one appears. In this case, the clock recovery process in the receiving SN will cause slide out of borders between LTOs of two SNs which communicate). This fact justifies our proposal for involving T_{ext} as time extension of T_{ON}.

For one-year period, the increase of *BC* in respect to its full capacity, when the duty cycle extension is implemented, is given in **Table 4**. In a concrete case duty-cycle extension is achieved by prolonging the time duration of a guard period, T_{guard}. For $T_{\Sigma} = 1$ s, and for *PRJR* in the range from 40% up to 70% the increment of *BC* is always less than 1.406%.

Table 4
Battery capacity increment *for one year period and different* σ *and PRJR.*

	σ =2	σ =3	σ =4	σ =5	σ =6	σ =7	σ =8
$PRJR_1 = 70\%$							
$T_{\Sigma} = 3600$ s	0.018	0.022	0.027	0.032	0.036	0.04	0.044
$T_{\Sigma} = 60$ s	0.364	0.461	0.558	0.655	0.75	0.84	0.925
$T_{\Sigma} = 1$ s	0.553	0.7	0.849	0.996	1.141	1.277	1.406
$PRJR_2 = 60\%$							
$T_{\Sigma} = 3600$ s	0.015	0.019	0.023	0.027	0.031	0.035	0.039
$T_{\Sigma} = 60$ s	0.313	0.396	0.479	0.563	0.644	0.721	0.794
$T_{\Sigma} = 1$ s	0.474	0.6	0.727	0.854	0.978	1.095	1.205
$PRJR_3 = 50\%$							
$T_{\Sigma} = 3600$ s	0.013	0.016	0.02	0.023	0.026	0.03	0.033
$T_{\Sigma} = 60$ s	0.261	0.33	0.4	0.47	0.538	0.602	0.663
$T_{\Sigma} = 1$ s	0.395	0.5	0.606	0.712	0.815	0.913	1.005
$PRJR_4 = 40\%$							
$T_{\Sigma} = 3600$ s	0.01	0.013	0.016	0.019	0.022	0.024	0.027
$T_{\Sigma} = 60$ s	0.209	0.265	0.321	0.377	0.431	0.483	0.532
$T_{\Sigma} = 1$ s	0.316	0.401	0.485	0.57	0.652	0.73	0.804

In case when we have several consecutively lost packets, by involving a duty cycling extension higher reliability in time synchronization is obtained. This is achieved at cost of very low increase of *BC* (*BC* < 1.406% per single year period for *PRJR* = 70% and σ = 8).

Diagrams that show the needed *BC* (for one year working SN period) in terms of different standard deviations starting from σ = 2 up to σ = 8, for PRJRs from 40% up to 70%, as parameters, and for IPI $T_{\sum} = 1$ s, are given in Fig. 6.

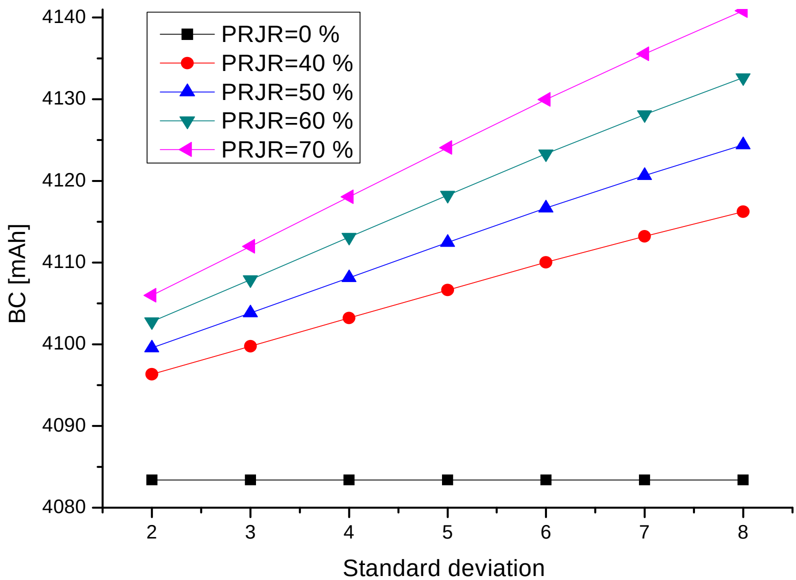

Fig. 6 – *Battery capacity for different PRJRs for single year SN's working period and* $T_{\sum} = 1$ s.

As we can see from Fig. 6, in all cases, the BC increases as σ increases. For example, in the worst case for *PRJR* = 70% (40%), σ = 2 (8), and $T_{\sum} = 1$ s, we obtain $\Delta BC_{70(40)} = 34.82$ mAh (19.91 mAh) what corresponds to relative *BC* increasment of 0.84% (0.48%). This implies that, in all cases, the *BC* increasement is minor, what justifies the involvement of T_{ext} in a guard period.

5 Conclusion

In almost all WSN applications, the achieved efficiency in power consumption is a very important issue. Several techniques to solve this problem have been developed. Here, we have described one of these techniques called duty-cycle extension. It relates to modification of a guard time, and is intended to compensate SN's local quartz frequency instability. The proposal allows us to achieve correct time synchronization in SN operation, even in a case when large number of consecutive packets is lost, and relatively high frequency instability (50 ppm) of the local SN oscillator exists. The described solution and its implementation are efficient. It does not require great resources. The obtained results show that, in the worst case, for single year working period, when the packet rejection ratio varies from 40% up to 70% and inter-packet interval is one second, an increase of battery capacity less than 1.406% is needed.

6 References

[1] M.K. Stojcev, M.R. Kosanovic, Lj.R. Golubovic: Power Management and Energy Harvesting Techniques for WSN, International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services, Niš, Serbia, 07 – 09 Oct. 2009, pp. 65 – 72.

[2] V. Raghunathan, S. Ganerival, M. Srivastava: Emerging Techniques for Long Lived Wireless Sensor Networks, IEEE Communication Magazine, Vol. 44, No. 4, April 2006, pp. 108 – 114.

[3] M.A.M. Vieira, C.N. Coelho, D.C. da Silva Jr., J.M. Mata: Survey on Wireless Sensor Network Device, IEEE Conference Emerging Technologies and Factory Automation, Lisbon, Portugal, 16 – 19 Sept. 2003, Vol. 1, pp. 537 – 544.

[4] T. Schmid, R. Shea, J. Friedman, Z. Charbiwala, M.B. Srivastava, H.C. Young: On the Interaction of Clocks and Power in Embedded Sensor Nodes, NESL Technical Report, University of California, Los Angeles, CA, USA, June 2009.

[5] M.R. Kosanovic, M.K. Stojcev: Delay Compensation Method for Time Synchronization in Wireless Sensor Networks, International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services, Niš, Serbia, 05 – 08 Oct. 2011, Vol. 2, pp. 623 – 629.

[6] I. Rhee, J. Lee, J. Kim, E. Serpedia, W.Y. Chung: Clock Synchronization in WSN: An Overview, Sensors, Vol. 9, No. 1, Jan. 2009, pp. 56 – 85.

[7] CC1000 Single Chip Very Low Power RF Transceiver, Texas Instruments, Dallas, TX, USA, 2009.

[8] CC2420 2.4 GHz IEEE 802.15.4/ZigBee-ready RF Transceiver, Texas Instruments, Dallas, TX, USA, 2013.

[9] K. Srinivasan, P. Dutta, A. Tavakoli, P. Levis: An Empirical Study of Low Power Wireless, ACM Transactions on Sensor Networks, Vol. 6, No. 2, Feb. 2010, Article No. 16.

[10] Z. Rosberg, R.P. Liu, T.L. Dinh, Y.F. Dong, S. Jha: Statistical Reliability for Energy Efficient Data Transport in Wireless Sensor Networks, Wireless Networks, Vol. 16, No. 7, Oct. 2010, pp. 1913 – 1927.

[11] P.K. Dutta, D.E. Culler: System Software Techniques for Low-power Operation in Wireless Sensor Networks, International Conference on Computer-Aided Design, San Jose, CA, USA, 06 – 10 Nov. 2005, pp. 925 – 932.

[12] S. Ganeriwal, D. Ganesan, H. Shim, V. Tsiatsis, M.B. Srivastava: Estimating Clock Uncertainty for Efficient Duty-cycling in Sensor Networks, International Conference on Embedded Networked Sensor Systems, San Diego, CA, USA, 02 – 04 Nov. 2005, pp. 130 – 141.

[13] T. Schmid: Time in Wireless Embedded Systems, PhD Thesis, University of California, Los Angeles, CA, USA, Sept. 2009.

[14] P. Dutta, D. Culler, S. Shenker: Procrastination Might Lead to a Longer and More Useful Life, Sixth Workshop on Hot Topics in Networks, Atlanta, GA, USA, 14 – 15 Nov. 2007.

[15] E. Jain, Q. Liang: Sensor Placement and Lifetime of Wireless Sensor Networks: Theory and Performance Analysis, Global Telecommunications Conference, St. Louis, MO, USA, 28. Nov. – 02 Dec. 2005, Vol. 1, pp. 173 – 177.

[16] H. Shu: Wireless Sensor Network Lifetime Analysis and Energy Efficient Techniques, PhD Thesis, The University of Texas at Arlington, Arlington, UT, USA, Aug. 2007.

[17] G. Pistoia: Battery Operated Devices and Systems, Elsevier, Amsterdam, The Netherlands.