FACTA UNIVERSITATIS
Series: **Electronics and Energetics** Vol. 28, N° 1, March 2015, pp. 85 - 102
DOI: 10.2298/FUEE1501085K

RPLL - RENDEZVOUS PROTOCOL FOR LONG-LIVING SENSOR NODE

Mirko R. Kosanović[1], Mile K. Stojčev[2]

[1]College of Applied Technical Sciences, Niš,
[2]University of Niš, Faculty of Electronic Engineering, Niš, Serbia

Abstract. *Sensor nodes (SNs), as constituents of wireless sensor network (WSN), are battery-powered not rechargeable devices and have limited amount of energy available. Since the lifetime of SNs is a crucial parameter for energy-efficient WSN design, it is essential to extend their lifetimes as much as possible. Here we propose a rendezvous scheme called Rendezvous Protocol for Long-Living SN, RPLL. This scheme is based on implementation of a duty-cycling technique. For each SN within WSN a unique identification number (ID) is allocated, thanks to which a collision problem is effectively remedied. The RPLL provides an on time wake-up of SNs in a fully decentralized way and fast detection of new appended SNs. Taking into account the WSN and SN working parameters, such as beacon time, beacon period, number of active SNs, and quartz oscillator instability, by using the proposed method, a WSN designer can determine the maximal lifetime of a SN, i.e. achieve optimal energy consumption.*

Key words: *rendezvous protocol; duty cycling; energy efficiency; wireless sensor networks*

1. INTRODUCTION

Wireless sensor networks (WSNs) consist of a large number of cooperating, radio-equipped and battery powered sensor nodes (SNs). Battery is the main power source in a SN. Bearing in mind that during its lifetime a SN operates with limited battery capacity (single battery charge) the energy consumption becomes a critical issue. For example, off-the-shelf SN works for a few days, if all its parts, including transceiver and microcontroller, are permanently powered-on. A SN is assumed to be dead when it is out of battery. The challenge is to guarantee lifetime of several years [1]. Four main activities during which a SN consumes energy are sensing, communication, computation, and storage [2]. The power consumed during communication is the greatest portion of energy consumption by any SN [1]. Communication between any two SNs is possible only if both of them are powered-on simultaneously. In order to arrange simultaneous on-time communication a rendezvous scheme is commonly used. Several ideas for rendezvous schemes in low duty cycle WSNs have been proposed [3]. They usually function at the

Received June 3, 2014; received in revised form August 25, 2014
Corresponding author: Mirko R. Kosanović
College of Applied Technical Sciences, Aleksandra Medvedeva 20, 18 000 Niš, Serbia
(e-mail: mirko.kosanovic@open.telekom.rs)

media access control (MAC) layer and can be categorized into three general classes [4, 5]: (i) asynchronous – the sender SN tries to capture the unknown active time of the receiver SN; (ii) synchronous – SNs are synchronized in time and agree on specific communications time slots; and (iii) pseudo asynchronous – SNs establish rendezvous on demand by using periodic wakeup. See Ref [4] for more details about this problematic. We consider a pseudo-asynchronous scheme because our application is primarily intended for rare events observation, i.e. in applications that are typical for environment monitoring. In this scheme SNs are powered on and off periodically, and a beaconing approach is used to express the desire or willingness to communicate. The concept of periodic on/off powering an SN, also called duty-cycling, has to satisfy performance requirements related to throughput, guaranteed end-to-end delay and a lifetime of several years, often contradictory design parameters. In general, duty cycling is the most widely used mechanism for saving energy in WSN, i.e. to elongate the WSN lifetime. The idea behind this is clear. Keep all or parts of hardware in low power sleep state except during instances when the hardware is operative. In this way, depending on the network activity, the SN switches its mode of operation between active and sleep. Duty-cycling protocols based on rendezvous scheme not only arrange for SNs to communicate, but also inherently include the availability to plan the channel access time, avoiding and resolving collisions, and some time-synchronization mechanisms which are required to locally determines the beginning of the active and sleep state [4, 6, 7].

Here we propose the usage of rendezvous scheme called Rendezvous Protocol for Long-Living SN, RPLL. In essence we continue our work [8], and present a complete RPLL protocol. The RPLL uses pseudo-asynchronous scheme and is based on the implementation of a duty cycling mechanism. Its principle of operation is similar to Distributed Low Duty Cycle (DLDC-MAC) protocol [9], from which we have adopted all advantages that it offers (synchronized wake-up times of SNs, hidden terminal, link failures, and asymmetric links elimination, etc.). With the aim to remedy the notified disadvantages of DLDC-MAC protocol, which primarily relate to collision avoidance during the registration of new SNs, as well as fast and correct selection of beacon time and beacon period, we propose its modification. The modification deals with involving a unique ID for each SN within WSN which provides us with integration of two activities (related to neighbor discovering and neighbor registration) into a single one (neighbor discovery and registration). This modification makes it possible to effectively overcome the aforementioned disadvantages. Our primary interest in this investigation is to determine how the working parameters of this protocol (beacon time, beacon period, number of communication active SNs, and quartz oscillator instability) have impact on power consumption, and to understand how to achieve a proper tradeoff between performance and power. To this end, the performances of RPLL protocol that relate to the number of active SNs, duration of a beacon period, and power consumption of SN, are estimated. For the defined and selected working parameters, by using the appropriate method, the designer can minimize the power consumption of SN.

2. Power Management Protocols and Energy Waste in WSNs

Depending on the layer on the network architecture they are implemented, power management protocols can be divided into the following two groups [10]:

a) Independent sleep/wakeup protocols - execute on top of a media access control (MAC) protocol. Since these protocols run at the network or application layer, they provide usage with any MAC protocol and are characterized with good adaptability to different application needs.
b) Strictly integrated with the MAC protocol - these protocols permit optimization of media access functions, but as specific solutions are not universal.

In terms of the approach which is used to determine when SNs should be switched-on, the sleep/wakeup protocols can be divided into the following three categories [5]:

1. On-demand – a SN should wake-up when another SN wants to communicate with it.
2. Scheduled-rendezvous – each SN should wake-up at the same time as its neighbors.
3. Asynchronous protocols – a SN should wake-up when it wants and still be able to communicate with its neighbors.

Depending on which way the source and destination SN achieve rendezvous, three categories of rendezvous schemes exist [11]:

1. Synchronous scheme – all SNs agree to the same clock time, wake-up synchronously and rendezvous with one another.
2. Asynchronous scheme – a source SN actively wakes-up destination SNs.
3. Pseudo-asynchronous scheme – a source SN wakes-up first and waits for destination SNs to wake-up and rendezvous.

As previously discussed, the energy is consumed by a SN for the sensing purpose, processing the data and communication. Communication consumes the largest amount of energy. In a typical SN, during communication, the major waste of energy occurs due to the following reasons [12]:

a) Idle listening – an SN carrier senses the idle channel in anticipation of possible arrival of packets, what causes waste of power.
b) Collision – when large number of SNs is present in a small area, collision is a common occurrence if it is not effectively controlled. A collided packet is discarded; an SN usually retransmits the packet, but packet retransmission causes a further waste of energy.
c) Overhearing - when SN’s neighbors are transmitting packets, although the packet is not designated for this SN, it still receives the packet, which represents yet another source of power waste.
d) Over-emitting – a SN sends packets to another SN but the receiver SN is not ready, and the packet has to be sent again, which also costs wasting power.
e) Control packet overhead – the presence of extra control packets in WSN such as a request to send (RTS), clear to send (CTS), acknowledgment (ACK), beacons, packets in CSMA-based protocols, and other transmitting and receiving control packets cost power as well.

3. Distributed Low Duty Cycle Rendezvous Protocol

First of all, we will analyze shortly the DLDC-MAC protocol, and point out to its principle of operation, advantages and drawbacks. The basic idea of this solution is described in [9]. For the sake of simplicity but without any intention to generalize the following discussion will explain DLDC-MAC protocol operation when the WSN consists of three SNs, SN_1, SN_2 and SN_3, respectively. Figure 1 presents a scenario of WSN activities which uses a DLDC-MAC protocol. As can be seen from Fig. 1 the following four activities (phases) exist:

1. Current State (***Cu_St***) – Each registered SN (in our case, SN_1 and SN_2) sends a short message called beacon periodically and wakes-up to receive beacons from its neighbors. The beacon period is the same for all SNs. After receiving a neighbor`s beacon the SN estimates the time of the next beacon taking into account the beacon period, the reception time, and the guard period. Having in mind that the SN executes this runtime for several neighbors it knows the beacon times for its neighbors. Thanks to this, the SN spends most of the time in a sleep state, and wakes-up only to receive neighbor`s beacons and to send its own beacon. Upon transmitting a beacon, each sending SN enters shortly into a receive mode during which it can accept data and commands piggybacked in the beacon. In this way each SN permanently takes an active part in communication and knows with how many SNs it communicates.

   Let us note that every phase (***Cu_St***, ***Ne_Di***, ***Ne_Re*** and ***Up_Sc***) is divided into several time slots denoted as $\boldsymbol{T_{BCY}}$ (see Fig. 1). Due to involving a guard time, used for compensation of SN`s quartz instability, each time slots is slightly longer than the time needed to send and to receive a beacon from its neighbor.

2. Neighbor Discovering (***Ne_Di***) – After powering-on (see Fig. 1.) the SN_3 enters into a phase ***Ne_Di*** and listens for a whole beacon period. During this period SN_3 receives neighbors` beacons from SN_1 and SN_2, and stores their reception times.

3. Neighbor Registration (***Ne_Re***) – After recognizing SN_1 and SN_2, the node SN_3 sends network join advertisement messages during the time periods when SN_1 and SN_2 are in the receive mode, respectively. The advertisement messages contain information which relate to the beacon time of the node SN_3.

4. Update Scheduler (***Up_Sc***) – This phase is identical to ***Cu_St*** phase with one exception, the joined SN_3 becomes active now, since SN_1 and SN_2 accept the beacons from SN_3.

The main advantages of DLDC-MAC protocol are:

i) Even in the presence of very unreliable links, it can successfully synchronize the wake-up times of SNs in a fully decentralized way.

ii) Problems accompanied with clock drifts, link failures, temporarily asymmetric links, and a hidden terminal, are effectively overcome.

iii) By using off-the-shelf SNs, long lifetime is possible to achieve.

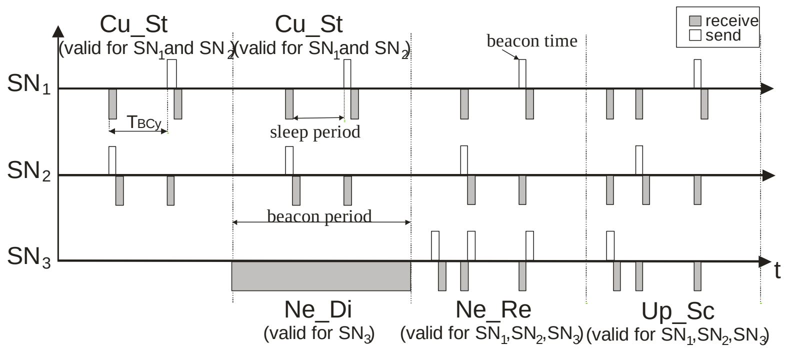


Fig. 1 DLDC-MAC rendezvous scheme

We meet the following drawbacks of DLDC-MAC protocol during:

a) Determination of an accurate beacon time during which the new joined SN tries to announce its presence in WSN, i.e. the selection of proper beacon time for the newly joined SN.
b) When the time difference between any two beacons is smaller than $\boldsymbol{T_{BCy}}$ (see Fig. 1), one of the affected SNs must choose a new beacon time. Since the beacon time is not pre-defined a problem arises when two or more new SNs choose the same beacon time, or the beacon time overlaps with the beacon period of other neighboring SNs during the new attempt period.
c) Collision appears when two or more SNs are simultaneously registered during the Ne_Re phase.

4. Rendezvous Protocol For Long-Living Sensor Node

In RPLL we assume that each SN is uniquely identified by its identification number, $ID_{\mathbf{x}}$, $\mathbf{x}$=1, 2 … $\boldsymbol{n}$, where $\boldsymbol{n}$ corresponds to the total number of SNs within WSN. Involvement of ID_x allows us to accurately define a unique time slot for each SN, within the beacon period, during which a corresponding SN can send data. Let us note that the unique ID is assigned during SN`s software initialization, i.e. in a phase of installing system and application software.

The scenario of events for RPLL is given in Fig. 2. The following three phases exist:

1. Current State (***Cu_St***) – identical to the one defined in DLDC-MAC.
2. Neighbor Discovering and Registration (***Ne_Di_Re***) – Since each SN has a unique ID_x it knows in advance the position of its time slot within beacon period. Each new joined SN during ***Ne_Di_Re*** phase enters in the receive mode. During a time slot T_3 (T_1), see Fig. 2, the node SN_2 accepts the beacon from SN_3 (SN_1) and notifies SN_3 (SN_1) about its presence in WSN, while during the time slot T_2 it announces its presence to the WSN (In this way SN_3 announces its presence to others SNs (for example SN_4, SN_5, …) as possible candidates that can attempt to join WSN during the same beacon period). This opportunity allows us to reduce a registration latency of the new joined SN (in Fig. 2 it is SN_2) to a single beacon period.
3. Update Scheduler (***Up_Sc***) – identical to that one defined in the DLDC-MAC protocol.

A procedure which deals with the joining of new SN to the WSN is presented in Figure 2 (in our case a new SN is SN_2). After powering-on, at the instant t_1, for a single beacon period $\boldsymbol{T_{BP}}$, SN_2 switches into the receive mode (***Ne_Di_Re*** phase) and accepts beacons from its neighbors. During this period each registered SN (SN_1 and SN_3) in a predefined time slot, determined by the SN`s ID number, sends a beacon. Since SN_2 accepts a beacon from its neighbor SN_3 (SN_1) it shortly switches to send a mode and announces a presence by sending its ID_2 to SN_3 (SN_1). In this moment SN_3 (SN_1) is in receive mode waiting to receive information from the new joined SN (in our case SN_2). According to the received ID_2 the node SN_3 (SN_1), in the next beacon period (during the phase Up_Sc) precisely determines a time slot for listening SN_2. In this manner, possible collision during registration of a new single joined SN (i.e. SN_2) is effectively avoided. However, let us note that a collision problem can appear when two or more SNs simultaneously join the WSN after powering-on during the same beacon period (for more details how to bypass this problem see subsection 4.1).

As can be seen from Fig. 2 the RPLL protocol differs in respect to DLDC_MAC (see Fig. 1) in that it merges Ne_Di and Ne_Re phases into a single phase called now Ne_Di_Re. The other two phases Cu_St and Up_Sc remain identical for both protocols. In this way, the needed latency for transition from phase Ne_Di to phase Cu_St (see Fig. 1) decreases for one beacon period. During Cu_St (Up_Sc) phase a beacon period always starts with sending beacon from SN with ID=1 (instant t_2 in Fig. 2). Thanks to this fact, it is easy to synchronize the operation, at the global level, of all SNs in WSN which relates to identification of the start and the end of each protocol phase.

Some important observations concerning the implementation of RPLL protocol are the following:

1. It is preferable to use RPLL protocol in WSN with a moderate number of SNs ($\boldsymbol{n} < 255$). In this case ID number is 8 bit width.
2. A designer assigns a unique ID number to each SN before its joining to WSN.
3. Time slot position of each beacon time, within a beacon period, is fixed and is directly determined by SN`s ID number.
4. Time duration of a beacon period is known in advance to each SN, i.e. defined during programming of a SN.
5. Minimal time duration of beacon period directly depends on a total number of SNs, $\boldsymbol{n}$, within a WSN.

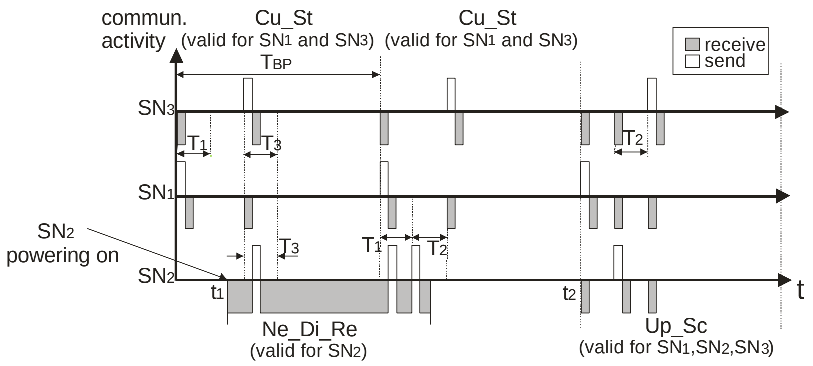


Fig. 2 Initialization phase without collisions.

4.1. Collision avoidance

Collision appears when during single beacon period several SNs are simultaneously switched on and try to transmit a packet at the same time. The following difficulty appears now in DLDC_MAC protocol: SNs that are in collision cannot accurately determine their beacon times, and cannot accurately define the duration of a total beacon period. Without loss of generality, in Fig. 3 we show how the collision problem in RPLL protocol can be solved. Namely, we assume that two sensor nodes SN_2 and SN_4 are powered-on during the same beacon period (i.e. time overlapping between phases ***Ne_Di_Re*** of SN_2 and SN_4 exists). During this we assume that $ID_1<ID_2< ID_3<ID_4$, where ID_1, ID_2, ID_3, and ID_4 are identification numbers of SN_1, SN_2, SN_3, and SN_4, respectively. As can be seen from Fig.3, at time period T_{RB1} (second beacon period in Fig. 3) collision appears during the phase ***Ne_Di_Re*** (for SN_2 and SN_4). A collision emerges because both SN_2 and SN_4 nodes after receiving a beacon from SN_1, during the period T_{RB1}, simultaneously shortly switch to send mode and announce their presences by sending their IDs to SN_1. At the end of a ***Ne_Di_Re*** phase (just before the start of Up_Sc [SN_2] phase) the following is known: a) All new joined SNs (SN_2 and SN_4) are aware about the presence of their neighboring SNs; b) The already active SNs (SN_1 and SN_3) have registered the newly joined SNs (SN_2 and SN_4) in a different way. Namely, SN_3 has registered the presence of all new SNs successfully, but for SN_1 it is not a case. As a consequence of collision SN_1 cannot register the presence of SN_2 and SN4. But now since the new joined SNs are aware that they are not registered by the already active SN_1, (acknowledgements are not received from SN_1) they activate their beacon planers. The beacon planer creates an ascending ordered list which contains IDs of unregistered SNs. This means that during the next phase marked us Up_Sc [SN_2] a node with a smaller ID (in our case it is SN_2 as the first item in beacon planer ordered list) will announce its presence to node SN_1, and will be joined to the WSN (becomes recognizable by SN_1) during Up_Sc [SN_4] phase. On the other hand, during a phase marked as Up_Sc[SN_4] a node with greater ID (in our case it is SN_4) will announce its presence to SN_1 and will be joined to WSN during a phase Cu_St. In this manner by using a beacon planer ordered list of ID items the collision problem can be effectively bypassed.

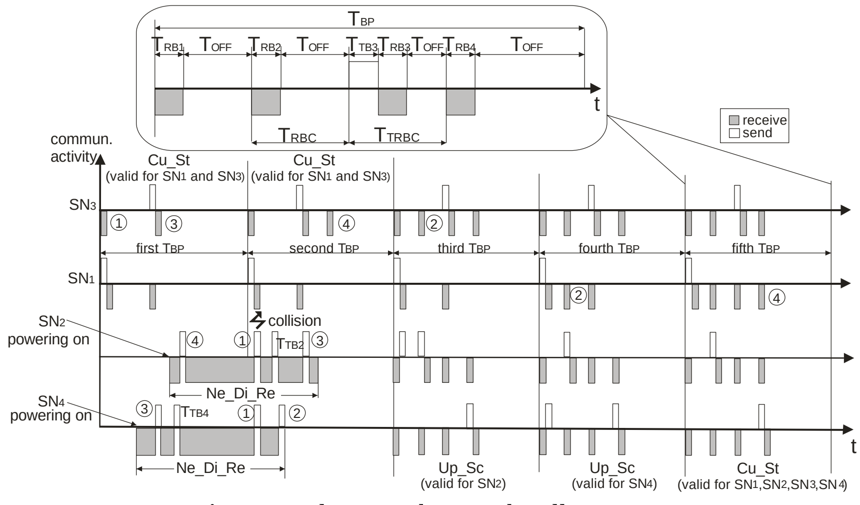


Fig. 3 Initialization phase with collision.

Notice: Symbols ①, ②, ③, and ④ deal with ordering of time slot position and instant of appearance within beacon period.

4.2. Time synchronization

Although all SNs run at same frequency, they all have a margin of errors. This means that they do not run at exactly the same speed, i.e. every clock will derivate from its intended nominal frequency for both dynamic (pressure, temperature, acceleration) and static (imprecision in its manufacture) reasons. This deviation is termed frequency error, and is defined as frequency drift. Due to frequency drift, a SN may wake up too soon or too early to communicate. To avoid this shortage we propose for each SN to piggyback its own local timestamp during transmitting its beacon. The receiving SN creates a local table in which each entry saves clock difference between the receiving and sending SN. Since most SNs in WSN have quartz oscillators with different stability, the receiving SN has a separate entry item in the table for all neighboring SNs. Similarly as in [9], during synchronization phase, these items are used as correction data for annulling the time difference between appropriate SNs. In this way, a unique time relation between each pair of neighboring SNs exists. This provides tight time synchronization between all pairs of neighboring SNs.

4.3. Link failures

In industrial environments where the level of electromagnetic disturbances is high, or links operate close to the noise floor, the packet rejection ratio is on average within the range from 40 % to 70 % [13]. Having this in mind, some modifications that relate to time synchronization, during the receiving process, are necessary to involve. 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, when several consecutive packets are not correctly received, the $\boldsymbol{T_{ON}}$ period can slide out of borders, within which it is expected to appear.

In the second solution, when the SN does not receive the packet in time, it updates its local time according to the value obtained during the last correctly received packet, and in additions it extends its guard time, $\boldsymbol{T_{guard}}$. In our design we adopt this approach. Let us note that the receiving SN has a separate item in its table for all neighboring SNs. Therefore, time corrections related to the guard time period are different [14].

5. Protocol Parameters Selection

In this section details which relate to the choice of working parameters such as duration of beacon period, determination of maximal number of communication active SNs, and duty cycle selection will be considered.

5.1. Duty cycle selection

From protocol point of view (see Fig. 3) most of its time (> 99.9 %) each SN spends into ***Cu_St*** or ***Up_Sc*** phase. During a single beacon period, $\boldsymbol{T_{BP}}$, there are maximum $\boldsymbol{n}$-1 receive beacon cycles, $\boldsymbol{T_{RBC}}$, and one transmit-receive beacon cycle, $\boldsymbol{T_{TRBC}}$.

For a given T_{BP} and defined duty cycle, DC, we will determine the maximal number of neighboring SNs with which some active SN can communicate. We assume that WSN consists of maximum n SNs. During ***Cu_St*** phase two different SN`s activities exist. The first one, with time duration T_{RBC}, is called receive beacon cycle. During this cycle the SN receives beacon from its neighboring SNs. From Fig.4a we have that $T_{RBC} = T_{RB} + T_{OFF}$. During T_{OFF} SN is in sleep mode, while in T_{RB} it receives beacon. The second activity, with time duration T_{TRBC}, is called transmit-receive beacon cycle. During this cycle, the SN transmits its beacon, and receives acknowledge from a new appending neighboring SN. From Fig.4b we have that $T_{TRBC} = T_{TB} + T_{RB} + T_{OFF}$, where T_{TB} corresponds to time period needed to acquire data from the sensor element and to transmit beacon. In our case we assume that $T_{RBC} = T_{TRBC} = T_{BC}$. T_{BC} corresponds to the duration of time slot defined in Fig. 2. For an arbitrary communication active (**com_active**) sensor node SN_k, k Є {1,..., n}, we define its duty cycle, DC_k (see Fig. 3 and Fig. 4), as:

$$DC_k = \frac{T_{ONk}}{T_{BP}} = \frac{T_{TBk} + T_{RBk} + \sum_{i=1}^{n} a_i T_{RBi}}{nT_{BC}} , \tag{1}$$

where the term a_i Є{0,1}, i=1,...,n, where i ≠ k, points to the following: a_i=1(a_i=0), a node SN_k can (cannot) communicate with its neighboring node SN, while T_{ONk} corresponds to the total active time of SN_k during a single beacon period T_{BP}. Within a WSN all active SNs are called **com_active**, while all SNs that directly communicate (point to point communication) are referred to as **com_visible**.
From eq. (1) we have,

$$T_{ONk} = T_{TBk} + T_{RBk} + \sum_{i=1}^{n} a_i T_{RBi} . \tag{2}$$

Let us assume now that during a ***Cu_St*** phase the number of **com_active** neighboring SNs is p, and that $p<n$ (for **com_active** SN_i the term a_i=1), so that the following is valid:

$$\sum_{i=1}^{n} a_i T_{RBi} = pT_{RB} . \tag{3}$$

If for some arbitrary **com_active** SN_k and **com_visible** SN_i we take that $T_{TBk} \approx T_{RBi} \approx T_{RB}$, for k Є{1, .., n}, i=1, .., n, then eq. (2) can be written as:

$$T_{ONk} = (p+2)T_{RB} . \tag{4}$$

By substituting eq.(4) into eq.(1) we obtain

$$DC_k = \frac{(p+2)T_{RB}}{T_{BP}} . \tag{5}$$

Activity profiles of the SN constituents (sensor, transceiver and CPU) during T_{RBC}, concerning time appearance and time durations, are sketched in Fig.4a. In Fig. 4b the activities of SN`s constituents during T_{TRBC} are presented. From Fig. 4a we have:

$$T_{RB} = T_{switch} + T_{guard} + T_{proc} + T_{guard} + T_{switch} \quad (6)$$

where T_{switch} is a CPU switching time between different operating modes (active and sleep); T_{proc} corresponds to CPU processing time; and T_{guard} represents a guard time.

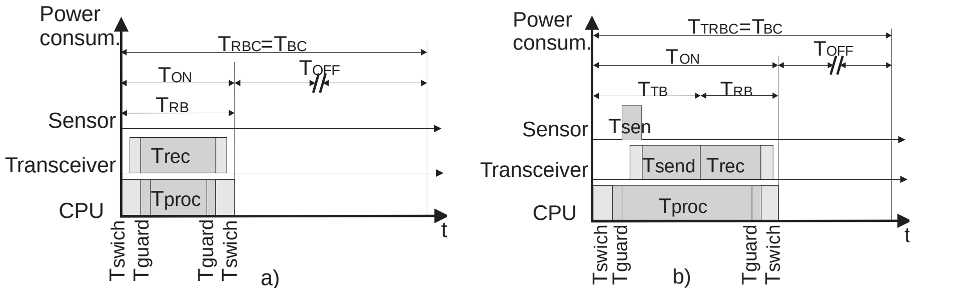


Fig. 4 Activity profile of SN during: a) receive beacon cycle; b) transmit-receive beacon cycle.
Notice: Let us note that T_{sen} stands for the needed time period for acquiring data from sensor element, T_{send} time period needed to sending beacon, T_{rec} time period needed for receiving data from new SNs, T_{proc} needed time periods for processing activity during T_{TRBC} and T_{RBC} beacon cycles.

During operation, the quartz instability results in relative clock drift appearance between SNs. As a result, nodes must include guard time. The guard time is equal to the maximum drift and linearly depends on T_{BP}. If quartz instability is denoted as $s_x(\Delta f/f)$, then the minimum guard time is:

$$T_{guard} = s_x T_{BP} \ . \quad (7)$$

By substituting eq.(6) into eq.(5) we obtain:

$$T_{BP} = \frac{(p+2)(2T_{switch} + T_{proc})}{DC_K - 2s_x(p+2)} \, . \quad (8)$$

Since T_{BP} must be always positive, i.e. $T_{BP}>0$, the following condition, related to eq. (8), has to be fulfilled:

$$DC_K - 2s_x(p+2) > 0 \, , \quad (9)$$

i.e. :

$$p < \frac{DC_K}{2s_x} - 2 \, . \quad (10)$$

According to eq. (10) the maximal number of **com_active** SNs within a single WSN region can now be derived. During this analysis, s_x will be taken as a parameter. In real WSN applications, the designer decides about: What kind of oscillator unit, in respect to quartz instability defined in ppm, to built-in within an SN structure? Usually, quartz units with factory declared quartz frequency instability from 10 ppm up to 50 ppm are built-in into SNs. Due to the process, voltage, and temperature (PVT) variations, as well as the influence of others ambient conditions (pressure, humidity, etc.), some additional frequency deviations, in respect to the factory declared quartz frequency instability, inevitably appear. We will consider a case when the impact of all additional frequency deviations is within the limits of ±10%. As direct consequence of frequency deviation, the number of ***com_active*** SNs will differ with respect to the number of SNs when quartz oscillators of nominally equal quartz frequency instability are used. In the worst case, for frequency deviation of +10 %, for fixed value of a DC factor, WSN with minimal possible ***com_active*** SNs is feasible. According to Eq. (10), for a given DC factor (from 0.1 up to 1 %), and specified frequency instability (from 10 up to 50 ppm) and frequency deviation of ±10 %, the maximal number of ***com_active*** SNs is presented in Table 1.

Table 1 Maximal number of ***com_active*** SNs in terms of DC factor

	10 (±10%)			**20**(±10%)			**30**(±10%)			**40**(±10%)			**50**(±10%)		
ppm / DC[%]	**-10%**	**nom**	**+10%**	**-10%**	**nom**	**+10%**	**-10%**	**nom**	**+10%**	**-10%**	**nom**	**+10%**	**-10%**	**nom**	**+10%**
0.1	52	47	42	24	22	19	15	13	12	10	9	8	8	7	6
0.2	108	97	87	52	47	42	34	30	27	24	22	19	19	17	15
0.3	163	147	133	80	72	65	52	47	42	38	34	31	30	27	24
0.4	219	197	178	108	97	87	71	63	57	52	47	42	41	37	33
0.5	274	247	224	135	122	110	89	80	72	66	59	53	52	47	42
0.6	330	297	269	163	147	133	108	97	87	80	72	65	63	57	51
0.8	441	397	360	219	197	178	145	130	118	108	97	87	85	77	69
1	552	497	451	274	247	224	182	163	148	135	122	110	108	97	87

*Note: In shaded columns the maximal number of ***com_active*** SNs within WSN, is derived, i.e. this column points to **p**.

By analyzing the results presented in Table 1 we can conclude that:

1. For the same DC factor the maximal number of ***com_active*** SNs always decreases as s_x increases, i.e. less stable oscillators are built-in.
2. Independently of quartz instability, as DC factor takes higher value, the maximal number of ***com_active*** SNs increases.
3. In all cases, small frequency deviation (from 9 up to 11 ppm) causes significant variation of ***com_active*** SNs. For example, for DC=1% and nominal s_x =10 ppm, the difference is 552-451=101 SNs.

5.2. Beacon period selection

For WSN with a maximal number of ***com_active*** SNs, let us determine now the beacon period, T_{BP}. To this end, in the sequel, two analyses related to the determination of a maximal number of ***com_active*** SNs within WSN, will be conducted. The first one deals with the choice of T_{BP} in terms of ***com_active*** SNs, for different quartz instability s_x and fixed (predefined) DC=0.5 %, as parameters. During the second analysis we determine T_{BP} duration in terms of ***com_active*** SNs for different DC factors (DC=0.1%, …, 1%) and fixed s_x (s_x =10ppm and s_x =40 ppm), as parameters, respectively.

First analysis: in general, low DC factor is preferable when long life SN operation is required. In practice the DC factor is within the range from 0.1 up to 1 % [15]. For illustration purpose only, we choose a middle value of DC (DC=0.5 %). In Fig.5, the minimal duration of T_{BP} in terms of the ***com_active*** SNs, with s_x as parameter, is sketched. During this we have adopted that:

a) $\boldsymbol{T_{proc}}$= 4 ms (in our case CPU runs at 1 MHz, packet length is 64 B, and data transfer rate is 128 kbps);
b) $\boldsymbol{T_{switch}}$= 6 µs (for a microcontroller of type MSP430F123, where 6 µs corresponds to a switching time from active to low power mode 3, and vice versa [16]).

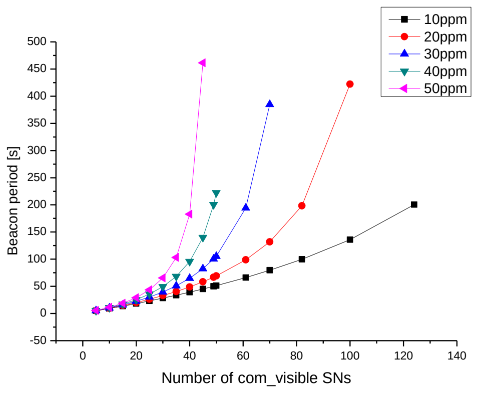
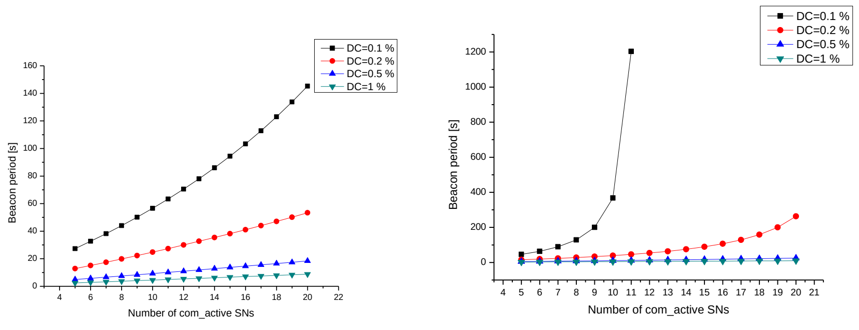


Fig. 5 Minimal duration of beacon period in terms of **`com_visible`** SNs for DC=0.5%

According to the results presented in Fig.5 we can conclude that a quartz oscillator instability has direct impact to the maximal number of **`com_visible`** SNs. Indenpendently of $\boldsymbol{T_{BP}}$, as $\boldsymbol{s_x}$ increases the maximal number of **`com_visible`** SNs decreases.

Second analysis: When DC factor is within the range from 0.1 up to 1 % similar results concerning minimal duration of $\boldsymbol{T_{BP}}$ are obtained.



a) b)

Fig. 6 Minimal duration of $\boldsymbol{T_{BP}}$ in terms of $\boldsymbol{p}$, for DC factor as parameter: a) $\boldsymbol{s_x}$=10 ppm; b) $\boldsymbol{s_x}$=40 ppm

Time duration of $\boldsymbol{T_{BP}}$ in terms of $\boldsymbol{p}$ ($\boldsymbol{p}$<20), for DC factor as a parameter, and for the fixed value of quartz oscillator instabillity $\boldsymbol{s_x}$ ($\boldsymbol{s_x}$=10ppm and $\boldsymbol{s_x}$=40 ppm), is sketched in Fig.6a and 6b, respectively.

By analyzing the results presented in Fig.6 we can conclude the following:

1. As DC factor decreases and the number of ***com_active*** SNs increases, T_{BP} increases, too.
2. As DC factor increases the curves defined by a function $T_{BP}=\Phi(p)$ become more linear.

According to the results presented in Fig.5 and Fig.6 we can conclude that: When quartz oscillator with lower ppm value is built into SN`s architecture, then:

a) For beacon period with fixed duration, WSN with larger number of ***com_active*** SNs can be made to be operative. For example, for T_{BP} =100 s and s_x =40 (10) ppm, WSN composed of p=35 (82) ***com_active*** SNs can be realized (see Fig. 5).

For WSN with a fixed number of ***com_active*** SNs, correct operation can be achieved with a lower DC factor. For example, for p=20 SNs, and s_x =50 (10) ppm, a DC factor of 1 (0,1) % is needed for feasible WSN operation (see Fig. 6).

6. Estimating Energy Consumption

It is well-known that battery factories declared that energy capacity, installed in SN, is not always equivalent to the energy drawn from that battery. With the aim to extract maximum energy from a battery, it is necessary to have a profound understanding of the following two phenomena. The first one deals with the amount of leakage current (the leakage current is a direct consequence of the battery self discharging characteristics). The second one relates to the energy consumption of SN during different phases of RPLL protocol. The energy consumption of SN is equal to:

$$E = \int_0^T I_{AVR}(t)Vdt + \int_0^T I_{SD}(t)Vdt = E_{SN} + E_{SD} \tag{11}$$

where E_{SN} is energy consumption of SN during its lifetime; E_{SD} is the wasted energy due to battery self discharge; V is power supply voltage; T lifetime of SN; I_{AVR} average current during lifetime of SN; and I_{SD} self discharge current.

With the aim to simplify the analysis we will assume that during lifetime of SN the power supply voltage is constant. In order to determine accurately the term $E_{SN}=I_{AVR}VT$ it is necessary to know the profile of I_{AVR}.

Current profiles of the sensor, transceiver, and CPU during phases ***Up_Sc*** and ***Ne_Di_Re***, are presented in Fig 7a and 7b, respectively.

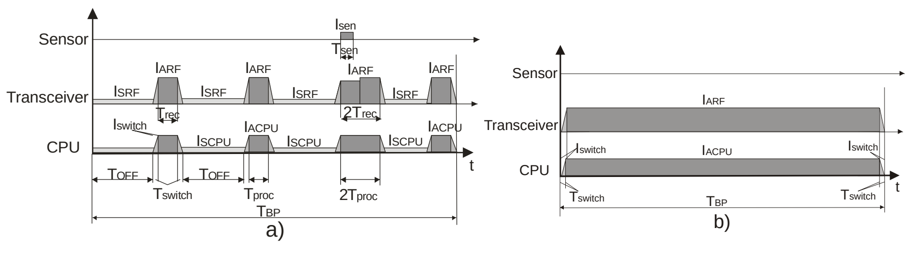


Fig. 7 Current profile of SN_3`s constituents during phases: a) ***Up_Sc***; b) ***Ne_Di_Re***

According to the principle of operation of RPLL protocol we have:

$$\mathrm{I}_{\mathrm{AVR}} = hI_{AU} + (1-h)I_{AD} + I_{SD} \tag{12}$$

where: $\boldsymbol{I_{AU}}$ - average current of SN during ***Up_Sc*** phase; $\boldsymbol{I_{AD}}$ – average current of SN during ***Ne_Di_Re*** phase; $\boldsymbol{I_{SD}}$ – battery self-discharging current; and, $\boldsymbol{h}$ – time ratio which points to the fact how long during its lifetime the SN spends into the phase ***Up_Sc*** in respect to ***Ne_Di_Re*** phase. Let us note that during the lifetime of SN the phase ***Ne_Di_Re*** happens at least once (during the registration of a new SN or after the reset), so limes $\boldsymbol{h} \rightarrow 1$.

The average current $\boldsymbol{I_{AU}}$ during the phase ***Up_Sc*** (see Fig.7a), is equal to:

$$\mathrm{I}_{\mathrm{AU}} = I_{AUCPU} + I_{AURF} + I_{AUsen} \tag{13}$$

where:

$$\mathrm{I}_{\mathrm{AUsen}} = I_{sen}\frac{T_{sen}}{T_{BP}}, \tag{14}$$

$$\mathrm{I}_{\mathrm{AURF}} = pI_{SRF}\frac{T_{OFF}}{T_{BP}} + 2pI_{switchRF}\frac{T_{switchRF}}{T_{BP}} + (p+1)I_{ARF}\frac{T_{rec}}{T_{BP}}, \tag{15}$$

and

$$\mathrm{I}_{\mathrm{AUCPU}} = pI_{SCPU}\frac{T_{OFF}}{T_{BP}} + 2pI_{switchCPU}\frac{T_{switchCPU}}{T_{BP}} + (p+1)I_{ACPU}\frac{T_{proc} + 2T_{guard}}{T_{BP}}. \tag{16}$$

The average current, $\boldsymbol{I_{AD}}$, during the phase ***Ne_Di_Re*** (see Fig.7b), is equal to:

$$I_{AD} = (I_{ACPU} + I_{ARF})\left(1 - \frac{2T_{switch}}{T_{BP}}\right) + 2I_{switch}\frac{T_{switch}}{T_{BP}} \tag{17}$$

From Fig. 7a we have that $\boldsymbol{I_{ACPU}}$, $\boldsymbol{I_{ARF}}$, $\boldsymbol{I_{sen}}$ correspond to the current values of CPU, transceiver, and sensor, respectively, when SN is in state on, while $\boldsymbol{I_{SCPU}}$, $\boldsymbol{I_{SRF}}$ to the current values of the CPU and transceiver during the time period when SN is in the state off.

In our case, for switching period $\boldsymbol{T_{switch}}$= 6 µs the CPU (MSP430F123) goes from active to LPM3 mode. Furthermore, we assume that during switching period $\boldsymbol{T_{switch}}$ the switching current $\boldsymbol{I_{switch}}$ varies linearly, and is given by the following formulas:

$$I_{switchRF} = \frac{(\mathrm{I}_{ARF} - I_{SRF})}{2} \tag{18}$$

$$I_{switchCPU} = \frac{(I_{ACPU} - I_{SCPU})}{2} \tag{19}$$

$$I_{switch} = \frac{(I_{ACPU} + I_{ARF} - I_{SCPU} - I_{SRF})}{2} \tag{20}$$

where: $\boldsymbol{I_{ACPU}}$ ($\boldsymbol{I_{SCPU}}$) and $\boldsymbol{I_{ARF}}$ ($\boldsymbol{I_{SRF}}$) correspond to CPU and RF currents during active (sleep) mode, respectively (see Fig.7).

By substituting eqs. (14), (15), (16) and (17) into eq. (12), the exact formula for calculating I_{AVR} is obtained.

The architecture of our SN consists of: a) CPU - low power microcontroller MSP430F123; b) communication part - RF modulator CC 2420; c) sensor subsystem - sensor MS55ER (for barometric-pressure); and d) Lithium-ion battery with capacity of 560 mAh. All presented results in the sequel relate to this SN architecture. In our design solution we have for I_{ACPU}=300 µA, I_{ARF}=19700 µA, I_{SCPU}=1.6 µA, I_{SRF}=1 µA, and I_{sen}=1000 µA. For rechargeable (non-rechargeable) lithium-ion battery, the self-discharging current, I_{SD}, causes battery capacity loss of 10 % (2%) per year [17].

The required battery capacity, RBC, for WSN composed of ten neighboring SNs (p=10), in terms of T_{BP}, for one year working period is sketched in Fig. 8. In Fig. 8, s_x is taken as a parameter, while the time ratio h = 0.9999884 corresponds to one discovery phase appearance during 24 hours time period.

As can be seen from Fig.8, in all cases, the minimal RBC is obtained for T_{BP}=60 s. Also, as the quality of quartz unit is better the RBC is lower. For example, for s_x=50 ppm the minimal RBC_{50}=670 mAh/year, while for s_x=10 ppm the minimal RBC_{10}=516 mAh/year, what corresponds to RBC increase of 23%. Under identical operating conditions for p=20, we obtain similar results (curves RBC=$\varphi(T_{BP})$ have nearly the same shape as those presented in Fig.8). As an illustration, for s_x=50 ppm and p=20, the minimal RBC_{50}=1188 mAh/year, while for s_x=10 ppm we obtain RBC_{10}=895 mAh/year, which corresponds to RBC increase of 25 %.

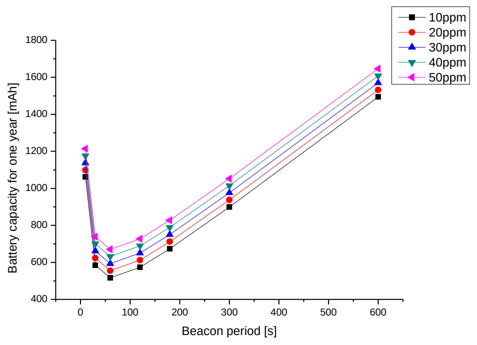


Fig. 8 Required battery capacity in terms of beacon periods with s_x as parameter

In Fig. 9 the power consumption of SN, for one year working period, in terms of T_{BP} with h as a parameter, is presented. According to the results presented in Fig. 9 we can conclude that:

a) As h decreases the power consumption of SN increases.
b) In all cases, minimal consumption exists, but for different T_{BP} values. For example for h=0.999988426 (single discovery cycle per day) and T_{BP}=60s we obtain E_{min}=670 mAh, while for h=0.9999996 (single discovery cycle per month) and T_{BP}=300 s, we obtain E_{min}=466 mAh, which corresponds to RBC increase of 30.45 %.

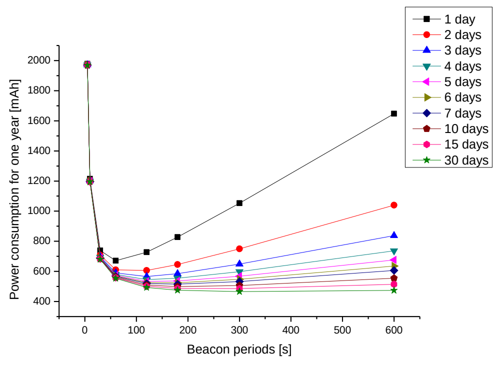


Fig. 9 Required battery capacity in terms of beacon period for different WSN scanning period

6.1. Power consumption comparison between DLDC-MAC and RPLL protocols

In Fig. 10 a power comparison between DLDC-MAC and RPLL protocol for one year period in terms of number of collisions, n_c, and time duration of beacon period, T_{BP}, as a parameter is given. The comparison is presented as ratio between SN`s power consumption of DLDC-MAC versus RPLL protocol, and is denoted as PC_R. By analyzing the results presented in Fig. 10 we can conclude the following:

i. Power consumption of DLDC-MAC is always higher in respect to RPLL protocol.
ii. For both protocols as n_c increases, the power consumption increases too. In addition, as n_c increases the PC_R increases too. For example: for T_{BP} = 180 s and n_c = 10 the PC_R = 2.81 %, while for T_{BP} = 180 s and n_c = 180 we obtain PC_R = 36.55 %.
iii. Higher power saving is always achieved for larger T_{BP}. As T_{BP} increases the PC_R increases too. For example: for T_{BP} = 180 s and n_c = 60 the PC_R = 15.95 %, while for T_{BP} = 600 s and n_c = 60 we obtain PC_R = 41.00 %.

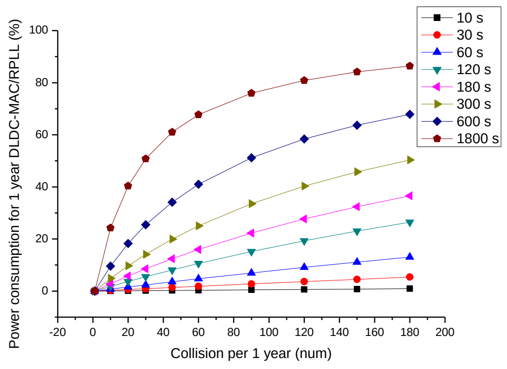


Fig. 10 Power comparison between DLDC-MAC and RPLL protocol for one year period in terms of number of collisions

The achieved saving in power consumption justifies the usage of RPLL in respect to DLDC-MAC protocol.

7. Conclusion

Power saving is a crucial issue in battery powered SNs. With the aim to achieve correct in-time wake-up of SNs we have implemented some modifications in respect to DLDC-MAC rendezvous protocol. The principle of operation of the proposed RPLL protocol is based on usage of duty cycling technique. It uses pseudo-asynchronous scheme and is preferable for WSN applications that are typical for environment monitoring. The modification in respect to DLDC-MAC protocol competes-in of involving a unique ID for each SN within WSN, which provides us with decreasing activities during the registration process and avoiding collisions of newly joined SNs. The performance of the proposed protocol which relate to: a) the maximal number of SNs within WSN in terms of a DC factor; b) the maximal duration of a beacon period in terms of simultaneously active SNs; c) the energy consumption of SN in terms of SN`s quartz oscillator instability; and d) the energy consumption of SN in terms WSN scanning period in order to detect newly active SNs, have been estimated. According to the obtained results, for a defined beacon period and selected SNs quartz oscillator instability, WSN designer can exactly determine the minimal power consumption of SN, and thus extend its lifetime. The power consumption of a SN which implements a RPLL protocol in respect to SN which uses DLDC-MAC protocol is always lower. The achieved power saving, for one year working period, is within the range from 0.01 % ($\boldsymbol{T_{BP}}$ = 1 s and $\boldsymbol{n_c}$ = 180) up to 86.43 % ($\boldsymbol{T_{BP}}$ =1800 s and $\boldsymbol{n_c}$ = 180). The obtained results justify the involved modifications in the RPLL protocol.

Acknowledgement: *This research was sponsored in part by the Serbian Ministry of Science and Technological Development, project no. TR-32009 - "Low-Power Reconfigurable Fault-Tolerant Platforms". The authors would like to thank to anonymous reviewers for their useful and constructive suggestions, mainly intended to improve the quality of this paper.*

References

[1] Raghunathan V., Ganerival S., Srivastava M., "Emerging techniques for long lived wireless sensor networks", *IEEE Communication Magazine*, Vol. 44, No. 4, pp. 108-114, 2006

[2] P. K. Dutta, D. E. Culler, "System Software Techniques for Low Power Operation in Wireless Sensor Networks", In Proceedings of the ICCAD`05, San Jose, California, USA, 2005, pp. 925-932

[3] M. Riduan Ahmad, Eryk Dutkiewicz and Xiaojing Huang (2011). "A Survey of Low Duty Cycle MAC Protocols in Wireless Sensor Networks", Emerging Communications for Wireless Sensor Networks, (Ed.), ISBN: 978-953-307-082-7, InTech, Available from: http://www.intechopen.com/books/emerging-communications-for-wirelesssensor-networks/a-survey-of-low-duty-cycle-mac-protocols-in-wireless-sensor-networks, acc.01.11.2013

[4] En Yi Lin, A Comprehensive Study of Power-Efficient Rendezvous Schemes for Wireless Sensor Networks, PhD thesis, University of California, Berkeley, 2005

[5] En-Yi A. Lin, Jan M. Rabaey, Adam Wolisz, "Power-Efficient Rendez-vous Schemes for Dense Wireless Sensor Networks", In Proceeding of ICC2004, Paris, France, June, 2004, Vol.7, pp. 3769 - 3776

[6] D. Christmann, R. Gotzhein, M. Krämer, M. Winkler, "Flexible and energy-efficient duty cycling in wireless networks with MacZ", Proc. 10th Annual Int New Technologies of Distributed Systems (NOTERE) Conf, IEEE, Tozeur, Tunisia, 2010, pp. 121-128

[7] E.Serpedin, Q.M.Chaudhari, Synchronization in WSN: Parameter Estimation, Performance Benchmarks and Protocols, Cambridge University Press, New York, 2009

[8] M.Kosanovic, M.Stojcev, "Energy Efficient Rendezvous Protocol for Wireless Sensor Networks, 2nd Mediterranean Conference on Embedded Computing", MECO – 2013, Budva, Montenegro, pp. 215-218

[9] M.Brzozowski, K.Piotrowski, P.Langendoerfer, "A Cross-layer approach for data replication and gathering in decentralized long-living wireless sensor networks", ISADS 2009, The 9th International Symposium on Autonomous Decentralized System, Athens 2009, pp.49-54

[10] Giuseppe Anastasi, Mario Di Francesco, Marco Conti, Andrea Passarella, *How to Prolong the Lifetime of Wireless Sensor Networks*, chapter 6 in Handbook on Mobile Ad Hoc and Pervasive Communications, L.T. Yang and M.K. Denko, Editors, American Scientific Publishers, December 2006, http://info.iet.unipi.it/~anastasi/papers/Yang.pdf, acc.10.11.2013

[11] G.Anastasi, M.Conti, M.D.Francesco, A.Passarelle, "Energy Conservation in Wireless Sensor Networks: A Survey", *Ad Hoc Networks* 7 (2009) 537–568, http://info.iet.unipi.it/~anastasi/papers/ adhoc08.pdf, acc.10.02.2013

[12] Wei Ye, Heidemann J., Estrin D., "An Energy-Efficient MAC Protocol for Wireless sensor networks", In Proceedings of IEEE INFOCOM 2002 (June 2002), New York, USA, Vol. 3, pp. 1567–1576

[13] Zvi Rosberg, Ren Ping Liu, Tuan Le Dinh, Yi Fei Dong, Sanjay Jha, "Statistical reliability for energy efficient data transport in wireless sensor networks", Wireless Netw (2010) 16, pp.1913-1927, Published on line by Springer Science + Business Media, http://www.cse.unsw.edu.au/~ydon/publications/winet. pdf, acc. 20.01.2013

[14] M. Kosanovic, M. Stojcev, "Delay Compensation Method for Time Synchronization in Wireless Sensor Networks",10 TELSIKS IEEE, Nis 2011, Vol.2 pp.623-629

[15] Anton Ageev, Time Synchronization and Energy Efficiency in Wireless Sensor Networks, DISI - University of Trento, Italy, PhD thesis, March 2010

[16] *Texas Instruments Datasheets*, MSP430F123 - 16-Bit Ultra-Low-Power Microcontroller, 8kB Flash, 256B RAM, USART, Comparator , http://www.ti.com/product/msp430f123, acc.15.02.2013

[17] Gianfranco Pistoia, *Battery Operated Devices and Systems*, Elsevier B.V., Amsterdam, The Netherlands, 2009