Novel Energy Consumption Model for Simulating Wireless Sensor Networks

Konstantin Mikhaylov and Jouni Tervonen
RFMedia Laboratory
Oulu Southern Institute, University of Oulu
Ylivieska, Finland
{konstantin.mikhaylov, jouni.tervonen}@oulu.fi

Abstract — **The simulation is nowadays one of the most widespread instruments for analyzing the Wireless Sensor Networks (WSNs). Unfortunately, the most widespread WSN simulators today have only the basic model for the battery, which does not account many of the important real-life hardware dependences. Therefore in the paper we are suggesting a novel approach to modeling the energy consumption for battery-powered WSN nodes. The suggested approach is based on the utilization of three dependences: the effect of battery charge on battery voltage; the effect of the supply voltage on the consumed by the WSN node current and the effect of the WSN node current on the available battery energy. The paper includes the description of the suggested module and its implementation in MiXiM framework for the OMNeT++ network simulator. Comparison of the results for real-life experiments, simulations using the standard approach and simulations utilizing the suggested approach are given. As revealed by the presented results, the suggested approach can significantly increase the simulation precision and provides additional functionality that can be valuable for WSN simulation.**

Keywords **- wireless sensor networks; WSN; simulation; battery; energy; model; consumption**

I. Introduction

The Wireless Sensor Networks (WSNs) have been under intensive research both by the academy and by the industry all over the world during the recent years. Over those years, numerous WSN applications have been developed and multiple real-life WSN deployments have been tested in-field. Although the real-life tests can provide valuable data, those require significant amount of efforts and resources and thus are often expedient only for the final stages of development [1], [2]. Besides, the results of the real-life tests are heavily affected by various environment parameters and interferences from the other systems. Therefore the simulation is nowadays one of the most widely adopted methods of analyzing WSN's behavior [3].

The existing WSN simulators can be classified into two major categories: the ones that extend the existing generic network simulators and the ones that have been designed specifically for the simulation of WSNs [3]. To the first group belong e.g., the extensions of NS (Network Simulator)-2[4] or NS-3[5], which are stated to be the most widely simulators for WSNs [1], [2], [6]. The NSs are the object-oriented discrete event network simulators written in C++ that originate from Real NS released in 1989 [3], [7]. SensorSim [8] and NRL [9] extend the functionality of NS-2 for WSNs simulations. Although the NSs are effectively extensible and there are multiple various protocols implemented for those, they are often criticized for the lack of the application model, the lack of customization and for the big difference between the used in NSs packet formats, energy models, MAC protocols, and the sensing hardware models and those describing typical real-life WSN platforms [1], [6], [7].

Another generic discrete event network simulating platform, which can be used for analyzing WSNs is OMNeT++ [10]. Like NSs, the models for OMNeT ++ are developed in C++, but OMNeT++ also uses NED language for specifying the network topology and INI files for defining the simulation parameters [1]. Based on OMNeT++, Castalia [11] and MiXiM [12] simulators, which are suitable for WSNs simulation, have been developed. The disadvantage of OMNeT++ and Castalia in specific is that those are not sensor-platform specific and lack the appropriate WSN sensing and energy consumption models [2], [7].

To the second group belong such simulators as TOSSIM [13] or COOJA [14]. TOSSIM is a bit-level discrete event simulator and emulator of TinyOS open-source operating system that uses in the simulations the actual application code [6], [13]. The major limitations of TOSSIM are that it works only with TinyOS-operated nodes, that it assumes each node in the network running the same code and that it lacks the energy consumption model although the last problem is partially solved in PowerTOSSIM extension [6], [7], [14]. COOJA simulator is a part of Contiki operating system that combines low-level simulation of sensor node hardware and simulation of high-level behavior [14]. Like TOSSIM, COOJA can be used both as a simulator and an emulator running - the actual application code e.g. for MSP430 lowpower microcontrollers that are often used on WSN nodes. Unlike TOSSIM, COOJA supports the networks with nodes having different on-board hardware and software although this requires sufficient resources and significant computing power [7].

It is well-known that the WSNs often contain the energy-constrained nodes that are powered from the batteries [3], [15]. This makes the energy one of the primary concerns in many WSN applications and boosts the need for the sufficient models of energy sources and energy consuming devices for accurate WSN simulations [16]. Nonetheless, the majority of the WSN simulators nowadays use only the linear battery model that is described by (1),

$$E = E' - P_i \cdot t \quad (1)$$

where E is the remaining battery energy after the consumption period t, E' is the remaining battery energy before the consumption period and P_i is power consumed for the activity i (e.g., radio packet transmission or receive, node sleep) over period t [3], [6], [7], [17–19]. In the majority of the existing WSN simulators the total available to the node energy is defined during the node initialization and does not depend on the current consumption profile. Similarly, the power P_i that the WSN node consumes for the activity i, is constant and does not depend on the battery status in most of the current simulations. Nonetheless, as has been shown in [20–23], these assumptions are far from truth for the real-life scenarios in case if the voltage stabilizing circuits are not used.

Therefore, in this paper we are suggesting the novel battery model and module implementing it in MiXiM framework. The suggested battery model provides additional information about the battery and helps to improve the precision for the WSN simulation.

II. Suggested Energy Model for WSN Simulations

As has been shown in multiple researches, the power consumption and the lifetime of the battery-powered device are affected by multiple parameters. The most important of those are: the supply voltage, the consumed by the WSN node current and current consumption profile [20–23]. In [21] the results of real-life experiments with the alkaline primary batteries have been presented, showing the discharge curves (i.e., the dependence on the battery's voltage from the time of operation or relative battery capacity) and the effect of the consumed current on available battery energy. As revealed in [20], [21], these dependences can vary significantly for different batteries and are often not trivial. Papers [20], [22–24] discuss the influence of the supply voltage level on the current, that is consumed by the WSN node's microcontroller and other peripherals. As has been revealed in those papers, the effect of supply voltage on the consumed current varies significantly depending on active WSN components and their operation modes.

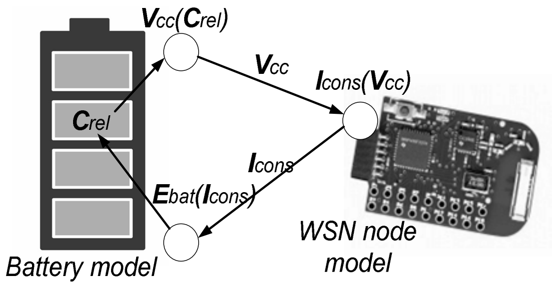

Figure 1. The major dependences for suggested WSN power source and consumption models.

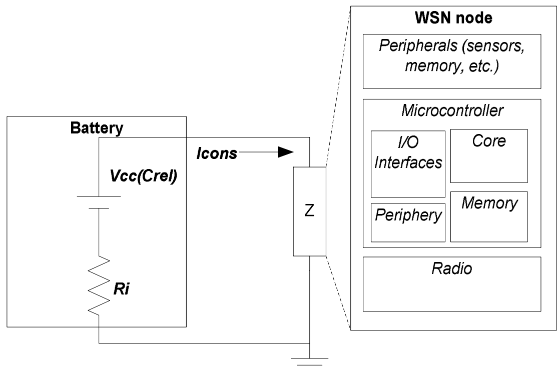

Figure 2. Simple equivalent circuit for the battery and the WSN node.

In order to take into account these dependences also during WSN simulations we are suggesting the following battery model (see Fig. 1). The available battery energy is represented as the relative battery capacity C_{rel} with possible value ranging from one (battery fully charged) to zero (battery discharged). The relative capacity can either decrease (e.g., if the current is consumed) or increase (e.g., due to the battery relaxation effect). Based on the value of C_{rel} at each stage of simulation is estimated the supplied by the battery voltage V_{cc}. This parameter can be used by the application model or models of other layers to estimate the possibility of launching the operations that require specific voltage levels (consider e.g., [20] for details). Besides, the V_{cc} in the suggested model is used for estimating the current for each WSN node's peripheral device and the cumulative consumed by the WSN node current I_{cons}. At the next step, I_{cons} is used for estimating the available for the battery nominal energy E_{bat}. Also, the value of I_{cons} can be used to specify the V_{cc} more precisely by taking into consideration the voltage drop on the battery internal resistance R_i (see Fig. 2 [25]). Therefore, the suggested battery model is described using (2).

$$Crel' = \frac{Ebat(Icons(Vcc(Crel))) \cdot Crel - (Vcc(Crel) - Icons(Vcc(Crel)) \cdot Ri) \cdot Icons(Vcc(Crel)) \cdot t}{Ebat(Icons(Vcc(Crel)))} \quad (2)$$

For the suggested model, we do not define the specific functions for describing either of $V_{cc}(C_{rel})$, $I_{cons}(V_{cc})$ or $E_{bat}(I_{cons})$ dependences. Instead, we suppose that these dependences are defined during battery model implementation based on the required simulation precision and available information about the battery and WSN node (consider Sections III and IV). Note, that for the case when $V_{cc}(C_{rel})$, $I_{cons}(V_{cc})$ and $E_{bat}(I_{cons})$ are defined as constants and $R_i=0$, the suggested battery model will be similar to the linear model described by (1).

III. Integration of Suggested Model into MiXiM

The standard framework for modeling the battery consumption in MiXiM is the Energy Framework presented in [18]. Nowadays (in MiXiM release 2.2.1) the only battery model available for Energy Framework is the *SimpleBattery* that provides the simple linear battery consumption model that is similar to (1) [18]. Therefore, the suggested in Section II model has been implemented as *AdvancedBattery* module for Energy Framework that can be used alternatively to the standard *SimpleBattery* model.

A. WSN battery consumption modelling in MiXiM

The Energy Framework in MiXiM has been designed to support multiple energy consuming devices on the WSN node [18]. These devices can represent the WSN node's microcontroller, the radio, the sensors/actuators or any other type of WSN node's peripherals. During the initialization of a device or in the case of the operation mode change, the module describing this device issues the special *draw* message that is forwarded to the battery module. The *draw* message contains the information on the new working mode of the device and the value of the current consumed by it in this mode. The battery module tracks the consumed by each WSN node's component current and uses these data for estimating the cumulative current consumption of the node.

Each time the battery module receives the new *draw* message; the battery module recalculates the consumed energy for the previous period, checks the status of the battery and estimates the available battery energy for the future. Besides, the battery module can use the special timer that forces the battery status update after the specified amount of time even if no new *draw* messages have been received.

Also, the battery module supports the public interfaces *estimateResidual()* and *getVoltage()* that can be used by any module to obtain the current value of the battery's voltage and battery's residual capacity [18].

The further details on Energy Framework for MiXiM architecture and operation can be found e.g., in [18].

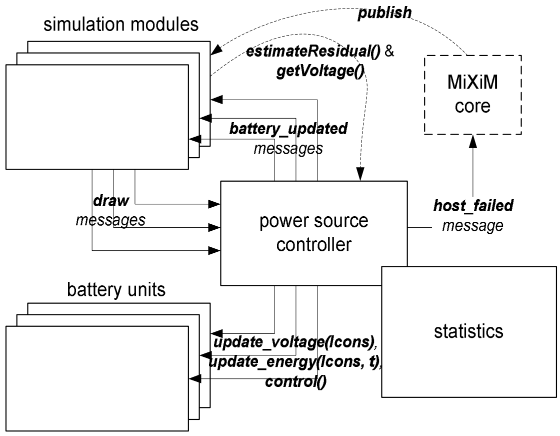

Figure 3. Architechture of the *AdvancedBattery* energy framework for MiXiM.

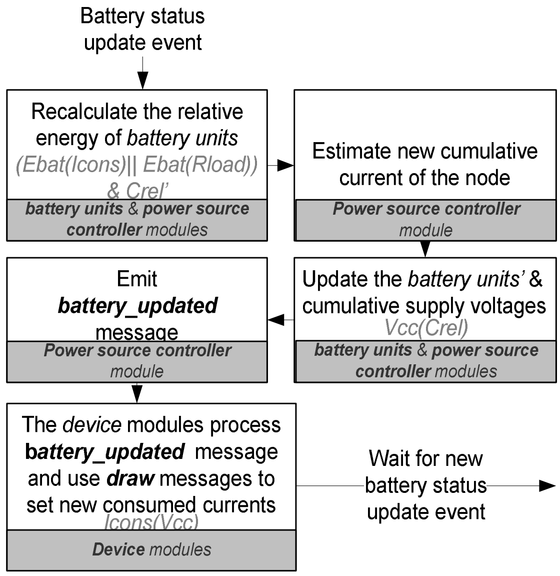

Figure 4. Update procedure for power source parameters in the developed modules and affected modules for each step.

One of the significant limitations for the current energy framework implementation in MiXiM is that the power source should be implemented within a single module and considered as whole regardless of the number of components (e.g., batteries). This approach makes it complicated to simulate e.g., the case when the node is supplied by several batteries with different characteristics.

As can be seen, for integration into Energy Framework for MiXiM the suggested in Section II energy model has to be separated into two major components. The first one is the power source model that accounts the battery relative capacity and introduces the $V_{cc}(C_{rel})$ and $E_{bat}(I_{cons})$ dependences. The second component is integrated into the modules describing the energy consuming devices, and defines the $I_{cons}(V_{cc})$ dependence for each particular device.

B. AdvancedBattery power source component implementation

For the sake of compatibility, we kept the general structure of the developed module similar to the one in the existing energy framework although several modifications were done (compare Fig. 3 with Fig. 1 in [18]). The first and the most significant one is that the module for the power source has been separated into two. The first one plays the role of *power source controller* and implements all the interfaces and message processing that are supported by the *Battery* module in standard framework as well as some new ones. The second module combines one or several sub-modules (so-called *battery units*), each describing a separate power source component (e.g., if the WSN node is powered by two AA batteries, each battery is described by the separate *battery unit*). The *power source controller* keeps the track and controls all the *battery units* (e.g., if required, it can "connect" or "disconnect" those from the WSN node). When the battery status should be updated (see Section III.A), the *power source controller* initiates the *update* procedure described in Fig.4.

As revealed in Figs. 3 and 4, one of the most important tasks for the *power source controller* unit is to track the cumulative voltage of all connected to the WSN node *battery units*. Besides, the *power source controller* unit implements the interface to the *statistics* module, which is responsible for logging the internal battery state. The *power source controller* unit is also responsible for detecting the power source failures. In the current implementation, a failure can be bind to two different events: either decrease of the cumulative *battery units'* energy (estimated for the currently used cumulative current value) or decrease of the cumulative supply voltage of the node's power source below the pre-defined threshold. In the case of node failure, the same mechanisms as in the standard energy framework of MiXiM are invoked [18].

As can be seen in Figs. 3 and 4, the *power source controller* does not make any calculations of the battery status on its own; instead it just issues the *update_voltage* or *update_energy* commands and provides the required input data for the *battery units*. The actual calculations of those parameters are made by the *battery units* and the dependences for $V_{cc}(C_{rel})$ and $E_{bat}(I_{cons})$ can be defined separately for each *battery unit*. In the case if the battery discharge data has been obtained not for the constant current but for constant load (e.g., like in [21]), instead of $E_{bat}(I_{cons})$ can be used $E_{bat}(R_{load})$ dependence. In the current implementation, either of $V_{cc}(C_{rel})$ or $E_{bat}(I_{cons})$/ $E_{bat}(R_{load})$ can be defined in three ways. The first option is the constant value. The second possibility is the linear function of the form $a \cdot x + b$. The final option that can be used for including real-life measurement results is to use the function defined as a table in the *.txt* file and linear approximation to the values between the table-defined points. The *battery unit* can be further extended with any other user-specified dependences for $V_{cc}(C_{rel})$ or $E_{bat}(I_{cons})$/$E_{bat}(R_{load})$. The required parameters for the functions (the value for the constant function, *a* and *b* values for linear and the input filename for the table-based functions) can be defined together with other simulation parameters in the *.ini* file for the simulation separately for each *battery unit*.

C. Devices model updates

The power source component, described in Section III.B, can be used also with the *device* modules developed for the standard energy framework of MiXiM. Nonetheless, in that case the simulation will not be able to account the changes of the consumed current due to the battery voltage decrease (i.e., $I_{cons}(V_{cc})$ dependence). To enable accounting it, the *device* modules should be updated: during the initialization phase the modules should subscribe to the special *battery_updated* message that is issued by the *power source controller* module once the cumulative WSN node's supply voltage has changed. Once such message is received, the *device* module processes it and estimates the novel value for the consumed current. In the implemented *device* modules, the same three possibilities for defining $I_{cons}(V_{cc})$, which were used for $V_{cc}(C_{rel})$ and $E_{bat}(I_{cons})$/$E_{bat}(R_{load})$, have been used. When the novel value of the current for the new voltage has been calculated, the *device* module issues the *draw* message to inform the *power source controller* about it.

IV. EVALUATION

In order to evaluate the suggested power model and to estimate the difference in the simulation results while using the existing and the suggested models we have simulated the trivial case – the device that was acting as a simple resistor (i.e., the consumed current was proportional to the voltage) has been connected to the module representing a single battery. For evaluating the efficiency of different models and for getting the required battery-related dependences we have used the results presented in [21]. The simulations have been done for six various models: Cn_Cn_Cn and Ca_Ca_Ca – the dependences (i.e., $V_{cc}(C_{rel})$, $E_{bat}(R_{load})$ and $I_{cons}(V_{cc})$) are modeled through constants equal to the nominal parameters (corresponds to the most widely used approach to modeling WSNs – i.e., (1)) in the first case and average values (based on experimental results from [21]) in the second case; L0_L0_L0, L1_L0_L0, Lt_Lt_L0 – all dependences are linear with parameters calculated using the experimental data in [21] (see below); PW_PW_L0 – all dependences are represented using the tables of experimental results. The dependences for $V_{cc}(C_{rel})$ and $E_{bat}(I_{cons})$ for different models for AG3 battery are illustrated in Figs. 5 and 6 respectively.

The $I_{cons}(V_{cc})$ for different models for the test bed are illustrated in Fig. 7. The results of the simulation revealing the example battery discharge curves and the lifetime of the system (estimated as time before the provided by the battery voltage reaches 0.8V

Table I Experimental results and results of simulation for "resistor" test bed using different models.

Battery type	Resistor, Ohm	Lifetime (experimental) ,s	Models											
			Cn_Cn_Cn		Ca_Ca_Ca		L0_L0_L0		L1_L0_L0		Lt_Lt_L0		PW_PW_L0	
			Lifetime,s	Error, %	Lifetime,s	Error, %	Lifetime,s	Error, %	Lifetime,s	Error, %	Lifetime,s	Error, %	Lifetime,s	Error, %
AG1	680	16137	21216	31	8736	-46	9596	-41	18398	14	15310	-5	19383	20
	150	2361	4680	98	1927	-18	579	-75	1072	-55	1138	-52	2498	6
	47	204	1466	619	604	196	88.1	-57	154.4	-24	220.5	8	177	-13
AG4	680	20956	29376	40	13585	-35	12584	-40	25193	20	20499	-2	24735	18
	150	2849	6480	127	2996	5	1097	-61	2171	-24	2077	-27	2907	2
	47	559	2030	263	939	68	241	-57	474	-15	502	-10	474	-15
AG3	680	81706	45696	-44	57446	-30	52011	-36	102147	25	81282	-1	94475	16
	150	10543	10080	-4	12672	20	6063	-42	12064	14	9389	-11	10959	4
	47	3297	3158	-4	3970	20	1570	-52	3142	-5	2422	-27	2822	-14
AG10	680	112228	130560	16	110935	-1	73220	-35	146034	30	111731	0	124954	11
	150	21903	28800	31	24471	12	13787	-37	25441	16	20532	-6	22764	4
	47	7701	9024	17	7667	0	4176	-46	7559	-2	6183	-20	6468	-16
AG13	680	245886	225216	-8	246440	0	158486	-36	315891	28	256617	4	289635	18
	150	51254	49680	-3	54361	6	26301	-49	52343	2	46154	-10	51325	0
	47	16652	15566	-7	17033	2	7713	-54	15345	-8	13825	-17	14097	-15
AAA	680	3068443	1632000	-47	2950290	-4	1874675	-39	3736845	22	2847633	-7	3059124	0
	150	599514	360000	-40	650797	9	359346	-40	704598	18	547885	-9	593202	-1
	47	176812	112800	-36	203916.6	15	109295	-38	213484	21	166783	-6	178355	1
RMS*				164.9		52.1		47.6		22.5		17.5		12

*- Root Mean Square (RMS) value of error for all tested scenarios

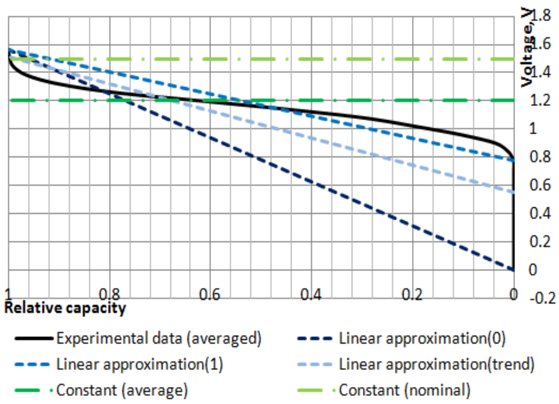

Figure 5. Illustration of $V_{cc}(C_{rel})$ dependence for various models of AG3 battery.

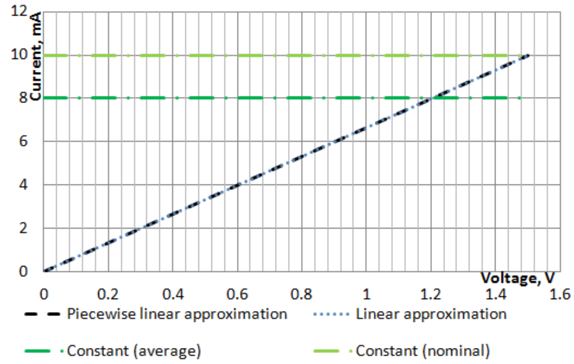

Figure 7. Illustration of $I_{cons}(V_{cc})$ dependence for various models of 150 Ohm "resistor" testbed.

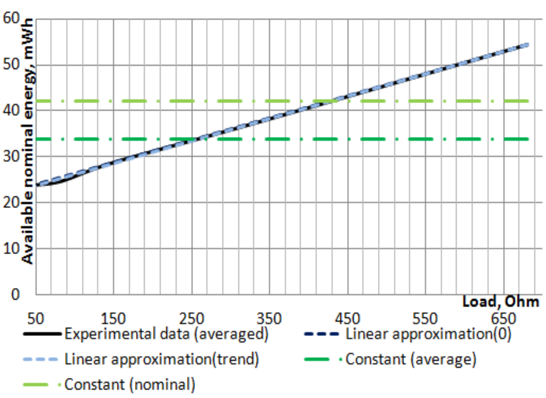

Figure 6. Illustration of $E_{bat}(R_{load})$ dependence for various models of AG3 battery.

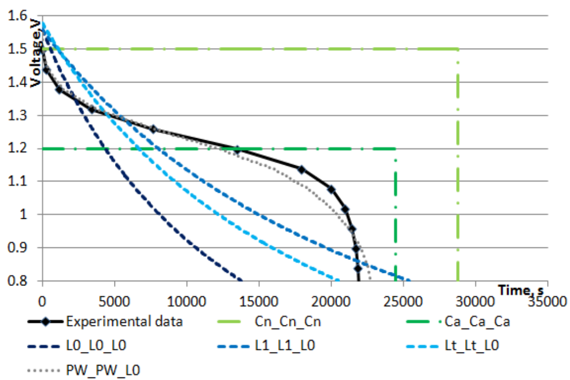

Figure 8. Experimental and simulated discharge curves for different models (150 Ohm load connected to single AG10 battery scenario).

threshold) are presented in Fig. 8 and Table I respectively.

As reveal the presented data, the use of the standard linear model (1) with the nominal parameters of the battery and nominal discharge current (Cn_Cn_Cn) for small-sized batteries (AG1 and AG4) with low capacity resulted in significant difference between the results of the real experiment and the simulations. Nonetheless, in some cases (e.g., for AG3 or AG13 batteries) the standard model provided quite good estimation of the lifetime. The use of the constant-based model with the parameters obtained in the experiment (Ca_Ca_Ca) allowed increasing the precision and reducing the total error. Nonetheless, as can be seen from Fig. 8, both those models do not consider the change of the battery voltage in any way.

As revealed in Table I, the L0_L0_L0 model sufficiently underestimates the lifetime of the system for each case. This model has been build under assumption that when the relative capacity of the battery is equal to zero the voltage is also zero (see Fig. 5) and assuming that there is linear dependence for $E_{bat}(R_{load})$ (see Fig. 6) and $I_{cons}(V_{cc})$ (see Fig. 7). The second linear model (L1_L0_L0) that assumes that battery "dies" when its voltage reaches 0.8 V provides better results although the mean percentage error (MPE) is still around 20%. The model that uses the trend-line for defining $V_{cc}(C_{rel})$ allows to reduce MPE even more (to 12%), although for some scenarios the error is still over 50%. Although the linear models try to introduce in simulation the voltage provided by the battery, as can be seen in Fig. 8 the resulting discharge curves still differ significantly from real experiment's results.

The last model that utilized the piece-wise approximation of the real-life experiment results allowed to decrease the maximum error for tested scenarios to 20% and MPE to 10%. As revealed in Fig. 8, the resulting discharge curve of the battery for this model is quite close to the experimental one.

During the development of the models, we were assuming that the $V_{cc}(C_{rel})$ is independent on the load and battery's discharge current. Unfortunately, that is not really so and this dependence has introduced some discrepancy between the experimental results and the simulations. In the simulation, we were also not considering the effects of internal battery resistance (it was assumed to be 0) and the recovery effect for the batteries, which also has played its negative role.

V. CONCLUSIONS

The simulation is today one of the most widespread instruments for analyzing the WSNs. Due to the wide utilization of the battery-powered nodes in WSN, the simulation of the energy and the energy consumption has significant importance during WSN simulation.

Therefore, in this paper we suggest the novel approach to modeling the energy consumption for battery-powered WSN nodes. Unlike the linear battery model that is most widely used in network simulators today, the suggested in the paper module tries to account during the simulation such effects as: the effect of the battery charge on the battery voltage $V_{cc}(C_{rel})$; the effect of the supply voltage on the consumed by WSN node current $I_{cons}(V_{cc})$ and the effect of the WSN node current on the available battery energy $E_{bat}(I_{cons})$. This approach allows one to introduce into the simulation the new parameters. One of those, which has significant influence on the operation of the WSN node as discussed e.g., in [20], [22], is the voltage supplied by the battery.

The suggested in the paper energy consumption modeling method has been implemented as a module for the energy framework for MiXiM extension of OMNeT++ network simulator. The implemented module has been tested with the three types of models (constant, linear or table-encapsulated experimental data) for each of the dependences $V_{cc}(C_{rel})$, $I_{cons}(V_{cc})$ and $E_{bat}(I_{cons})$. As reveal the presented results, in some scenarios the models using constant dependences can have very significant errors for defining the lifetime of the device. The linear dependences provide somewhat better results although in some cases the error is still above 50%. Nonetheless, the discharge curves obtained using the linear models still differ significantly from the real-life ones. The experiment-based dependences allowed to reduce the error for lifetime estimation below 20% and to get the discharge curves that are close to real-life ones.

One of the significant benefits of the developed energy module comparing with the existing one is that WSN node's power source can be represented by the combination of several independent units (e.g., separate unit for each battery) which are handled separately. This allows one to test more complicated energy control techniques and scenarios during the simulation.

In the future, we are planning to extend the suggested models through also considering the effects of the battery internal impedance, the recovery effects and the effect of the environment parameters (e.g., the temperature) and test the developed energy models for more complicated scenarios (e.g., the real WSN node operation and real WSN scenario).

REFERENCES

[1] M. Jevtic, N. Zogovic, and G. Dimic, "Evaluation of Wireless Sensor Network Simulators," in *Proceedings of TELFOR'09*, Belgrade, Serbia, pp. 1303 –1306.

[2] A. Rastegarnia and V. Solouk, "Performance evaluation of Castalia Wireless Sensor Network simulator," in *Procedings of TSP'11*, Rome, Italy, 2011, pp. 111 –115.

[3] G. V. Merrett, N. M. White, N. R. Harris, and B. M. Al-Hashimi, "Energy-aware simulation for wireless sensor networks," in *Proceedings of SECON'09*, Piscataway, NJ, USA, 2009, pp. 64–71.

[4] ISI, University of Southern California, CA, USA, "The Network Simulator - ns-2." [Online]. Available: http://www.isi.edu/nsnam/ns/. [Accessed: 25-Apr-2012].

[5] NSNAM, "What is ns-3." [Online]. Available: http://www.nsnam.org/overview/what-is-ns-3/. [Accessed: 25-Apr-2012].

[6] M. Korkalainen, M. Sallinen, N. Karkkainen, and P. Tukeva, "Survey of Wireless Sensor Networks Simulation Tools for Demanding Applications," in *Procedings of ICNS '09*, Arlington, VA, USA, 2009, pp. 102 –106.

[7] H. Sundani, H. Li, V. Devabhaktuni, M. Alam, and P. Bhattacharya, "Wireless Sensor Network Simulators A Survey and Comparisons," *IJCN*, vol. 2, no. 6, pp. 249 –265, 2011.

[8] S. Park, A. Savvides, and M. B. Srivastava, "SensorSim: a simulation framework for sensor networks," in *Procedings of MSWIM'00*, New York, NY, USA, 2000, pp. 104–111.

[9] I. Downard, "Simulating Sensor Networks in NS-2." [Online]. Available: http://cs.itd.nrl.navy.mil/pubs/docs/nrlsensorsim04.pdf. [Accessed: 25-Apr-2012].

[10] A. Varga and R. Hornig, “An overview of the OMNeT++ simulation environment,” in *Proceedings of Simutools’08*, Brussels, Belgium, 2008, pp. 60:1–60:10.

[11] A. Boulis, “Castalia, a simulator for wireless sensor networks and body area networks, version 3.2.”

[12] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin, “Simulating wireless and mobile networks in OMNeT++ the MiXiM vision,” in *Procedings of Simutools’08*, Brussels, Belgium, 2008, pp. 71:1–71:8.

[13] P. Levis, N. Lee, M. Welsh, and D. Culler, “TOSSIM: accurate and scalable simulation of entire TinyOS applications,” in *Procedings of SenSys’03*, New York, NY, USA, 2003, pp. 126–137.

[14] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, “Cross-Level Sensor Network Simulation with COOJA,” in *Proceedings of LCN’06*, Tampa, FL, USA, 2006, pp. 641 –648.

[15] J. Eriksson, “Detailed Simulation of Heterogeneous Wireless Sensor Networks,” Dissertation for the degree of Licentiate of Philosophy in Computer Science, University of Uppsala, Uppsala, Sweden, 2009.

[16] E. Egea-López, A. Vales-Alonso, A. Martínez-Sala, P. Pavón-Mariño, and J. García-Haro, “Simulation Tools for Wireless Sensor Networks,” in *Procedings of SPECTS’05*, Philadelphia, PA, USA, pp. 2–9.

[17] G. Chen, J. Branch, M. J. Pflug, L. Zhu, and K. Szymanski, “Chapter 1 SENSE: A SENSOR NETWORK SIMULATOR,” *Components*, pp. 249–267, 2004.

[18] L. M. Feeney and D. Willkomm, “Energy Framework: an extensible framework for simulating battery consumption in wireless networks,” in *Proceedings of the Simutools ’10*, Brussels, Belgium, 2010, pp. 20:1–20:4.

[19] A. Stetsko, M. Stehlik, and V. Matyas, “Calibrating and Comparing Simulators for Wireless Sensor Networks,” in *Proceedings of MASS’11*, Valencia, Spain, 2011, pp. 733 –738.

[20] K. Mikhaylov, J. Tervonen, and D. Fadeev, “Development of Energy Efficiency Aware Applications Using Commercial Low Power Embedded Systems,” in *Embedded Systems - Theory and Design Methodology*, K. Tanaka, Ed. Rijeka, Croatia: InTech, 2012, pp. 407 –430.

[21] K. Mikhaylov and J. Tervonen, “Experimental Evaluation of Alkaline Batteries’s Capacity for Low Power Consuming Applications,” in *Procedings of AINA’12*, Fukuoka, Japan, 2012, pp. 331 –337.

[22] K. Mikhaylov and J. Tervonen, “Optimization of microcontroller hardware parameters for Wireless Sensor Network node power consumption and lifetime improvement,” in *Procedings of ICUMT’10*, Moscow, Russia, 2010, pp. 1150 –1156.

[23] D. Raskovic and D. Giessel, “Dynamic Voltage and Frequency Scaling For On-Demand Performance and Availability of Biomedical Embedded Systems,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 6, pp. 903 –909, Nov. 2009.

[24] C. Seem, “Programming Output Power on CC243x,” Texas Instruments, Design Note SWRA171, 2007.

[25] D. I. Pomerantz, “The characterization of high rate batteries,” *IEEE Transactions on Consumer Electronics*, vol. 36, no. 4, pp. 954 –958, Nov. 1990.