Accurate Online Energy Consumption Estimation of IoT Devices using Energest

Adnan Sabovic, Carmen Delgado, Jan Bauwens, Eli De Poorter, Jeroen Famaey

Abstract

Minimizing the energy consumption of Internet of Things (IoT) devices is one of the biggest challenges and crucial issues for the future of a sustainable IoT vision. In order to estimate the remaining device lifetime and optimize its energy consumption, it is necessary to have an accurate online view on the consumed energy with minimal overhead. This is non-trivial, as many factors influence energy consumption, therefore requiring a generic measurement methodology. For example, the Medium Access Control (MAC) protocols have a very important influence on the energy consumption. This paper presents an accurate method for estimating the energy consumption of IoT devices using Energest. Our method combines a device-specific offline profiling phase, with a device and protocol-agnostic online energy estimation methodology. Energy measurements have been performed for different scenarios, using measured values and values from the datasheet, for Carrier Sense Multiple Access (CSMA) and Time Slotted Channel Hopping (TSCH) protocols. Results show that the accuracy of our method is very high, more than 96% for CSMA and more than 82% for TSCH, with very small overhead of 0.11%.

Key words: IoT, Energest, energy estimation, Zolertia RE-Mote, CSMA, TSCH

Adnan Sabovic, Carmen Delgado, Jeroen Famaey
IDLab, University of Antwerp – imec, Antwerp, Belgium,e-mail: (adnan.sabovic, carmen.delgado, jeroen.famaey)@uantwerpen.be
Jan Bauwens, Eli De Poorter
IDLab, Ghent University – imec, Ghent, Belgium,e-mail: (jan.bauwens2, eli.depoorter)@ugent.be

1 Introduction

Internet of Things (IoT) is a paradigm used to connect objects to the Internet, where billions of devices cooperate and communicate with each other with the aim to simplify and improve daily life. To achieve the full potential of the IoT, it is necessary to develop new systems and technologies that are able to minimize energy consumption, as most of these objects are battery powered. Low-Power Wireless Sensor Networks (LPWSNs) are presented as one of the best possible solutions that combine low power consumption with long-range communications.

Since IoT devices are usually equipped with small batteries, that are expensive and short-lived, the reduction of their energy consumption is one of the biggest challenges and crucial issues for the future of a sustainable IoT vision. Energy efficiency has become one of the key criteria for designing Wireless Sensor Networks (WSNs), first because of the ecological aspect, but also to ensure the functionality of the sensor nodes for a long period without recharging or replacing batteries. The network's lifetime directly depends on the energy consumption of sensor nodes and minimizing and modeling of energy consumption are the main objectives for designing energy-efficient WSNs [4].

Several mechanisms to measure and estimate energy consumption online have been proposed that are based on the accumulated time values of MCU and radio usage. For example, Energest is a software-based energy estimation mechanism implemented as a Contiki NG [1] module, that provides functions to measure the accumulated time the sensor nodes spend in different MCU (active, low power, etc.) and radio (TX, RX, LISTEN, OFF, etc.) states [9].

In order to optimize the lifetime of IoT devices, it is necessary to get an accurate view on their consumed energy while they are online. This would allow battery lifetime prediction, energy-aware operations, etc. Traditionally, Energest, and other similar frameworks, rely on datasheet energy consumption values to transform timing information into energy consumption estimates. We show this leads to inaccurate results, and instead propose a generic device profiling methodology.

In this paper, we propose an accurate method for estimating the energy consumption of IoT devices using Energest. Through our experiments, we show the accuracy of our method, where we combine Energest with real power consumption values from a power analyzer. We also evaluate the overhead of using Energest in terms of energy consumption.

2 Related work

WSNs are composed of a large number of interconnected sensor nodes which usually use batteries as their main source of energy. Therefore, the design and devel-

[1] https://github.com/contiki-ng/contiki-ng

opment of low-power and energy-efficient WSNs has become a major challenge for the new wave of IoT technologies.

Power consumption of a sensor node depends on many factors, such as the states of the radio, microcontroller (MCU), Light-emitting diodes (LEDs) and other components [10]. Different mechanisms have been used before to estimate energy consumption on small IoT devices such as Contiki-NG's built-in funcionality, named Energest [2]. It is a time-based estimation mechanism implemented as a collection of functions and macros that runs directly on the sensor nodes and measures the accumulated time the sensor node spends in different MCU and radio states [10][9].

Time-based energy estimation is very easy to implement and configure with existing applications, but the accuracy of the obtained results is not always good enough. Possible problems about the accuracy of the results are shown by Steinfeld et al. [10]. For example, the CPU mode is activated whenever the node is active, including the period when the radio transmits or receives some data, but not for all the time needed for a transmission. After it makes the radio ready to transmit data, the MCU can switch to the LPM mode. It is very hard to know the accurate period of the CPU and the LPM state of the MCU during one transmission and almost impossible to estimate the energy consumption of the MCU and the radio separately. Steinfeld et al. [10] propose a hardware-based approach, named Smart Coulomb Counter (SCC), that easily adds to a sensor node the capability of measuring its energy consumption. Also, they compare their solution with Energest and get very similar results between SCC and Energest in the OFF state, but in the ON state, the consumption values differ greatly (more than 85% of relative error) because of the LED current consumption. In contrast, we use Energest to get time values and estimate the energy consumption using actual measured current values. This greatly improves the accuracy without a need for expensive hardware.

Daneels et al. [6] present an accurate energy consumption model for devices, using both the 868MHz and 2.4GHz frequency bands, using the TSCH protocol. They identify all network-related CPU and radio state changes, providing a precise representation of the device behavior and an accurate prediction of its energy consumption. Their approach is protocol-specific, while we propose a general methodology. There are a few works where Energest has been used as the main software for calculating power consumption. For example, Schandy et al. [9] presented a simple approach for the analysis of the average power consumption of a sensor node according to the node states and network protocols that have been used. Kharce and Pawar [8] analyzed the energy consumption in a 6LoWPAN network for real and emulated Zolertia Z1 motes using the Energest-based module Powertrace to record each of the possible states of the radio.

Some authors such as Dunkels et al. [7] and Wu et. al. [12], used Powertrace to estimate the total energy consumption of the system. Powertrace tracks the state of the device to estimate the power consumption for individual tasks that are captured in energy capsules. Powertrace is based on Energest and records the energy consumption of the activity by opening an energy capsule when the activity starts

[2] https://github.com/contiki-ng/contiki-ng/wiki/Documentation:-Energest

and closing when it ends [7]. Wu et. al. [12] used values from Powertrace and the datasheet of the MCU and radio to get the energy consumption from every possible state. Hussain et al. [5] measured the energy consumption for different states of the radio and the MCU depending on the number of nodes in the network using Powertrace with datasheet values, and performing simulations for more network scenarios. Both Energest and Powertrace can estimate the energy consumption of the radio and the MCU, by recording how long each component spends in a specific state.

In contrast to these existing Energest-based methodologies, we use more accurate device profiling, rather than datasheet values. Moreover, we show that our method is agnostic to the used MAC protocol, and evaluate its overhead.

3 Methodology

In this section, the basics of the used software, hardware, MAC protocols and how our measurements have been performed are described.

3.1 Software

3.1.1 Contiki NG

Contiki-NG[3] is an open-source, cross-platform operating system for the next generation of IoT devices that is focused on enabling reliable and secure low-power communication using standard protocols, such as Internet Protocol version 6 (IPv6), Constrained Application Protocol (CoAP) or IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL). This is a new version of the Contiki project that provides an RFC-compliant, low-power IPv6 communication stack and enables Internet connectivity [2]. Contiki-NG can be used for building a lot of new WSN programs and it supports many hardware platforms, such as Zolertia Zoul, Tmote Sky/TelosB, OpenMote cc2538, etc.

3.1.2 Energest

Energest[4] is a lightweight, software-based energy estimation mechanism for resource-constrained IoT devices that provides functions to measure the time the device is in different states. This is a Contiki-NG's built-in functionality implemented as a collection of macros and functions. The macros are used to tell the Energest module when a component changes its mode or to return the total time the Energest mod-

[3] https://github.com/contiki-ng/contiki-ng

[4] https://github.com/contiki-ng/contiki-ng/wiki/Documentation:-Energest

ule has been tracking. The functions are usually used to initialize Energest, update the total time for all types that are currently turned on or to return the total time for the specified state of the device. There are five Energest modes that all Contiki-NG platforms support: CPU, LPM, Deep LPM for the microcontroller (MCU) and TRANSMIT, LISTEN, OFF for radio. Not all MCUs support both LPM and Deep LPM and in that case, the unused type will always report 0 seconds as time. Note that RECEIVE is not considered as a separate radio state, and reported as LISTEN.

3.2 Hardware

3.2.1 Zolertia RE-Mote platform

The Zoul[5] is a core module developed by Zolertia[6] that provides a flexible and compatible module solution to be integrated with most existing products and solutions. The Zoul module can be used for industrial and IoT projects, to ease the development of new WSNs and applications for them, from intelligent lighting systems to monitoring applications in Smart Homes and Cities.

The Zolertia RE-Mote[7] is an ultra-low power complete hardware development platform, based on the Zoul, designed jointly by universities and industrial partners, easy and flexible to use for most of IoT applications. The platform is based on the Texas Instruments CC2538 ARM Cortex-M3 System on Chip (SoC)[8], with an on-board 2.4 GHz IEEE 802.15.4 RF interface. It supports two radios and it is compatible with existing and trending protocols, for both indoor and long-range IoT applications with maximum range between 100m and 20km. Contiki-NG has been successfully ported to the Zolertia RE-Mote platform.

3.2.2 N6705B DC Power Analyzer

The N6705B DC Power Analyzer[9] is a device that provides the possibility for sourcing and measuring voltage and current levels of different devices. The N6705B offers flexible configuration to meet power sourcing and analysis requirements. It integrates capabilities of up to four advanced power supplies with Digital Multimeter (DMM), Scope, Arb and Data Logger features. All functions and measurements are available at the front panel and there is no need for developing or debugging programs to control something on the instrument.

[5] https://github.com/Zolertia/Resources/wiki/The-Zoul-module

[6] https://zolertia.io/

[7] https://github.com/Zolertia/Resources/wiki/RE-Mote

[8] http://www.ti.com/product/CC2538

[9] https://www.keysight.com

3.3 MAC protocols

To show that our methodology is MAC-protocol-agnostic, we evaluated it with two commonly used MAC protocols for WSN and IoT applications as described below.

3.3.1 CSMA protocol

CSMA is a network protocol that listens to the channel before transmitting to check if it is idle or not [3]. When the channel is detected idle for transmission, the device can send the packet. Otherwise, it will perform a random back-off before retrying. Only one device can send packets through the channel, otherwise a collision will occur and the transmission will fail. The radio is listening to the channel all the time and never goes to the OFF state, while the MCU allows the Low Power Mode (LPM) state, but with the highest level of power consumption.

Figure 1 illustrates the channel states for the CSMA protocol for one transmission between two nodes, transmitter and receiver, for a random period. Different states of the radio (LISTEN, TX and RX) and MCU (CPU and LPM) are shown.

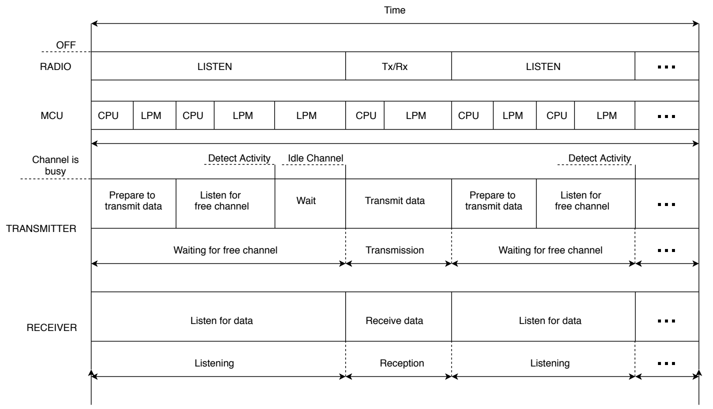


Fig. 1: Channel state for CSMA protocol without ACK mechanism for a transmitter and receiver node

3.3.2 TSCH protocol

TSCH mode, specified in IEEE 802.15.4e [1], is an ultra-low power and highly reliable medium access control technology [11]. It aims to achieve a high reliability of 99.99% and minimal power consumption, which are very important for industrial and other challenging IoT environments [6]. In TSCH networks, all nodes are synchronized following a time-synchronized schedule, divided into time slots of generally 10 or 15ms, which instructs every node exactly when to send or receive data and how to avoid wasting valuable energy. Moreover nodes hop between the available channels in a pseudo-random manner to avoid interference. The sender's and receiver's radio are ON only during their assigned slots. After that, both nodes can switch their radios to OFF and go to the sleep mode for few milliseconds before repeating the same procedure in order to transmit/receive an acknowledgment (ACK) [11].

A timeslot template for a transmitter and receiver node, with different MCU (CPU, LPM, and Deep LPM) and radio states (LISTEN, Tx, Rx and OFF), is shown in Figure 2. Radio and MCU spend most of the time in OFF and Deep LPM states, as such saving a large amount of energy.

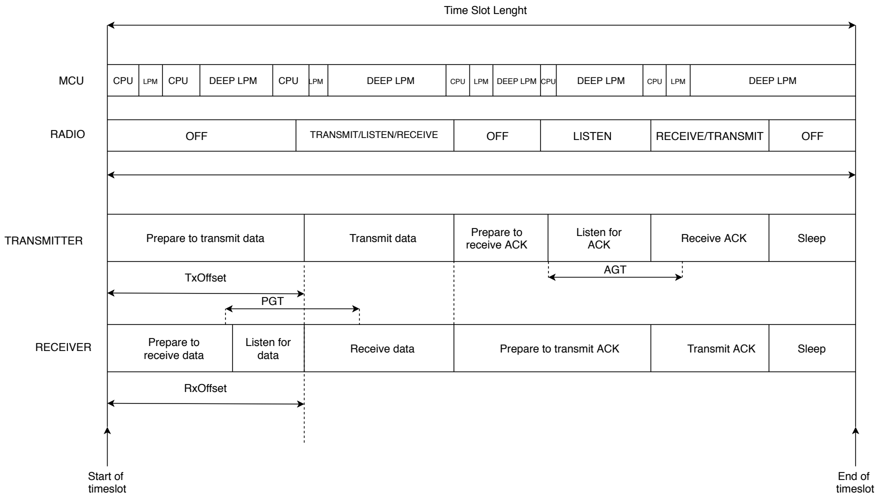


Fig. 2: TSCH time slot template for a transmitter and receiver node

3.4 Measurement methodology

For energy estimation, we used Energest as a timing mechanism together with energy consumption values that have been measured with the power analyzer. We have performed four types of measurements with two different MAC protocols, CSMA and TSCH. The energy consumption E_{tot} in Joule is calculated as follows:

$$E_{tot} = \sum_{s \in S} E_s = \sum_{s \in S} t_s \times I_s \times V \tag{1}$$

where S is the set of MCU and radio states, t_s is the time spent in the state s in seconds (as reported by Energest), I_s is the current consumption of s in Ampere and V is the operating voltage in Volts.

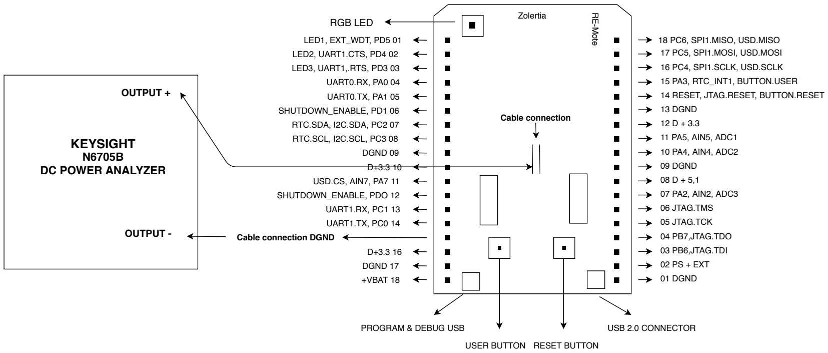


Fig. 3: Energy consumption measurement setup

The measurement setup is shown in Figure 3. In order to increase the accuracy of the energy estimations, we have measured the energy consumption of the Zolertia RE-Mote MCU and radio states with the power analyzer. We removed the resistor to enable the direct voltage supply of 3.3V for the chip (MCU and radio) which decreased the total energy consumption of the device. As such, our methodology consists of using measured current values for each I_s rather than values obtained from datasheets.

4 Results

4.1 Experimental setup

We have created an application for two Zolertia RE-Mote devices that operate in the 2.4GHz frequency band using the CC2538 radio. The duration of our experiments is 60 seconds with packet transmission interval of 5 seconds, unless stated otherwise. Two different packet sizes have been tested, 8 and 64 bytes. We performed the experiments using two different MAC protocols, CSMA and TSCH. For all the cases, we used the same operating voltage of 3.3V.

We repeated each experiment ten times and our results are averages over these iterations. As a baseline, we compare to Energest with datasheet values and use the power analyzer to measure the total device current consumption over the entire experiment duration to get the real consumed energy.

4.2 Device profiling

Table 1 shows the differences between measured and datasheet values for different MCU and radio states. As expected, the measured current consumption is much higher for the LPM and Deep LPM states compared to the datasheet. This can be explained because there are other components that are affecting energy consumption, which are not taken into account in the datasheet values. In Section 4.3 we use both the datasheet and profiled values for I_s in combination with Energest.

Table 1: Comparison between values from the datasheet and N6705B DC Power Analyzer

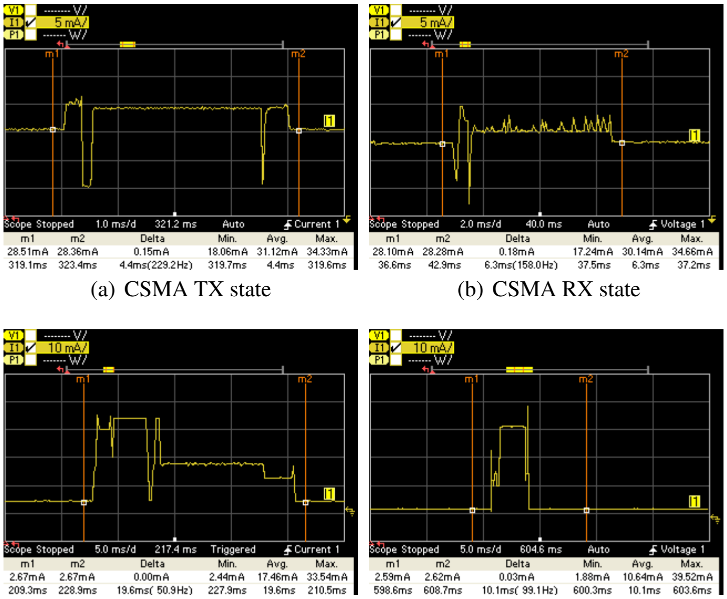
State	**CC2538 datasheet**	**Device profiling**
CPU	20 mA	15.35 mA
LPM	0.6 mA	9.59 mA
Deep LPM	0.0013mA	2.58 mA
LISTEN	24 mA	28.32 mA
Rx	27 mA	30.14 mA
Tx	34 mA	31.12 mA

Figure 4 shows the current drawn for the TX and RX states using the CSMA protocol and the current drawn for TX and RX time slots using TSCH as MAC protocol, both when using the CC2538 radio.

4.3 Accuracy of Energest

Table 2 shows the average power consumption of the Zolertia RE-Mote plaform for the two different MAC protocols and for two different packet sizes. The first column (Datasheet + Energest) has been calculated using Equation 1, using for I_s the current consumption from the datasheet. For the second scenario, we combined Energest with measured values from the power analyzer to obtain more accurate values for I_s. The two last experiments show the actual total energy consumed by the device, with Energest active and inactive respectively. It shows that the impact of Energest is 0.11%, which means that it doesn't consume a lot of energy.

Table 2 shows the accuracy of using Energest with datasheet values and real measured values for different MCU and radio states. In case of CSMA, the accuracy of using Energest with datasheet values is 91% for both an 8B and 64B payload. When using real measured values with Energest instead the datasheet, the accuracy increases to more than 96%, which is a significant improvement. For TSCH, the accuracy of using Energest with datasheet values for an 8B and 64B payload is less than 53%. But, when using real measured values the accuracy increases to more than 82% for both payloads.



Scope Stopped 1.0 ms/d 321.2 ms Auto Current 1

m1	m2	Delta	Min.	Avg.	Max.
28.51mA	28.36mA	0.15mA	18.06mA	31.12mA	34.33mA
319.1ms	323.4ms	4.4ms(229.2Hz)	319.7ms	4.4ms	319.6ms

Scope Stopped 2.0 ms/d 40.0 ms Auto Voltage 1

m1	m2	Delta	Min.	Avg.	Max.
28.10mA	28.28mA	0.18mA	17.24mA	30.14mA	34.66mA
36.6ms	42.9ms	6.3ms(158.0Hz)	37.5ms	6.3ms	37.2ms

(a) CSMA TX state

(b) CSMA RX state

Scope Stopped 5.0 ms/d 217.4 ms Triggered Current 1

m1	m2	Delta	Min.	Avg.	Max.
2.67mA	2.67mA	0.00mA	2.44mA	17.46mA	33.54mA
209.3ms	228.9ms	19.6ms(50.9Hz)	227.9ms	19.6ms	210.5ms

Scope Stopped 5.0 ms/d 604.6 ms Auto Voltage 1

m1	m2	Delta	Min.	Avg.	Max.
2.59mA	2.62mA	0.03mA	1.88mA	10.64mA	39.52mA
598.6ms	608.7ms	10.1ms(99.1Hz)	600.3ms	10.1ms	603.6ms

(c) TSCH TX time slot

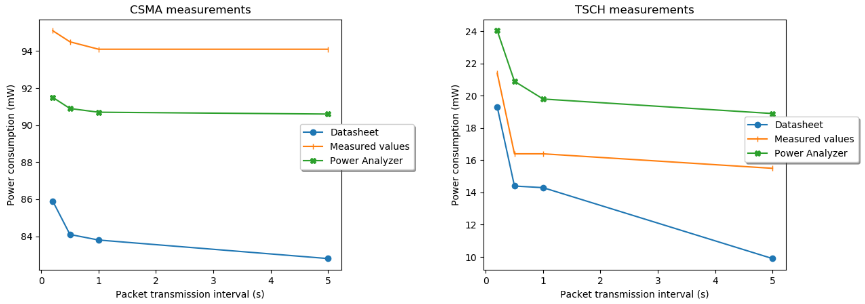
(d) TSCH RX time slot

Fig. 4: Measured current consumption for TX and RX states using CSMA and TSCH protocol

Table 2: Power consumption of Zolertia RE-Mote platform

Node	MAC model	Packet size	Datasheet + Energest	Real Values + Energest	Power Analyzer + Energest ON	Power Analyzer + Energest OFF
Transmitter	CSMA	8B	82.7 mW	93.9 mW	90.7 mW	90.6 mW
Receiver			82.7 mW	93.9 mW	90.5 mW	90.5 mW
Transmitter	CSMA	64B	82.8 mW	94.1 mW	90.8 mW	90.6 mW
Receiver			82.7 mW	93.9 mW	90.6 mW	90.6 mW
Transmitter	TSCH	8B	9.8 mW	15.5 mW	18.7 mW	18.7 mW
Receiver			9.7 mW	15.4 mW	17.9 mW	17.8 mW
Transmitter	TSCH	64B	9.9 mW	15.5 mW	18.9 mW	18.9 mW
Receiver			9.8 mW	15.4 mW	18.3 mW	18.2 mW

The device consumes more energy when using the CSMA protocol compared to TSCH, with differences between the protocols of more than 70mW. Using CSMA, the device never goes to the Deep LPM state, but spends most of the time in the LISTEN state using a lot of energy. In contrast to CSMA, when using TSCH, the MCU spends most of the time in the Deep LPM state, so it saves a large amount of energy. Also, the radio turns off when the device is not in a transmission or reception slot, which also significantly reduces the power consumption of the device. The two last experiments show the impact of Energest that is 0.11% which means that it doesn't consume a lot of energy.



(a) CSMA measurements

(b) TSCH measurements

Fig. 5: Power consumption for different packet transmission intervals for 64 bytes

We measured the power consumption for different packet transmission intervals i.e. 0.2, 0.5, 1 and 5 seconds, for both used protocols, as shown in Figure 5. Different packet transmission intervals also have an impact on energy consumption, as they affect the number of transmissions. The results also show that when using measured current values the accuracy of Energest with actual measured values is not affected by the transmission interval. However, when using datasheet values it is, and accuracy decreases as the transmission interval grows.

5 Conclusions

In this paper, an accurate method for energy estimation of Zolertia RE-Mote devices using Energest has been presented. We have analyzed four different scenarios for different MAC protocols (CSMA and TSCH) and different packet sizes (8 and 64 bytes). We used current consumption values from the datasheet, but also measured values for every possible state of the device. Our results showed that the values used from the datasheet are not accurate and the current consumption for some states of the device is much higher because of the impact of other device components. We have shown that the power consumption depends on the used MAC protocol, packet size, device configuration and packet transmission interval. Our results show that using accurate state current measurements, Energest has an accuracy of more than 96% for CSMA, and more than 82% for TSCH. Thus is a great improvement compared to the use of datasheet values, where the accuracy is 91% for CSMA, and less than 53% for TSCH. Moreover, Energest has an overhead in terms of power consumption of only 0.11%, making it on highly suitable for online energy estimation in low-power IoT devices.

Acknowledgment

Part of this research was funded by the Flemish FWO SBO S004017N IDEAL-IoT (Intelligent DEnse and Long range IoT networks) project.

References

1. IEEE standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (LR-WPANs) amendment 1: Mac sublayer. IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011) pp. 1–225 (2012). DOI 10.1109/IEEESTD.2012.6185525
2. Practical Contiki-NG. Apress (2018)
3. Abdulrazzak, H.N., Hussein, A.A.: Performance analysis of CSMA/CA based on D2D communication. International Journal of Computer Sciences and Engineering (2019)
4. Abo-Zahhad, M., Farrag, M., Ali, A.: Modeling and minimization of energy consumption in wireless sensor networks. In: 2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS), pp. 697–700 (2015). DOI 10.1109/ICECS.2015.7440412
5. Ahmad, H., Ali, J., Waqar Shah, S., Nayab, D.: Optimizing computations in intermittently powered wireless sensor nodes. International Journal of Engineering Works (2018)
6. Daneels, G., Municio, E., Van de Velde, B., Ergeerts, G., Weyn, M., Latre, S., Famaey, J.: Accurate energy consumption modeling of IEEE 802.15.4e TSCH using dual-band OpenMote hardware. Sensors (2018)
7. Dunkels, A., Eriksson, J., Finne, N., Tsiftes, N.: Powertrace: Network-level power profiling for low-power wireless networks. SICS Technical Report (2011)
8. Kharche, S., Pawar, S.: Node level energy consumption analysis in 6LoWPAN network using real and emulated zolertia z1 motes. In: 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), pp. 1–5 (2016). DOI 10.1109/ANTS.2016.7947870
9. Schandy, J., Steinfeld, L., Silveira, F.: Average power consumption breakdown of wireless sensor network nodes using IPv6 over LLNs. In: 2015 International Conference on Distributed Computing in Sensor Systems, pp. 242–247 (2015). DOI 10.1109/DCOSS.2015.37
10. Steinfeld, L., Oreggioni, J., Bouvier, D., A. Fernandez, C., Villaverde, J.: Smart coulomb counter for self-metering wireless sensor nodes consumption. Journal of Low Power Electronics (2015)
11. Vilajosana, X., Wang, Q., Chraim, F., Watteyne, T., Chang, T., Pister, K.S.J.: A realistic energy consumption model for TSCH networks. IEEE Sensors Journal **14**(2), 482–489 (2014). DOI 10.1109/JSEN.2013.2285411
12. Wu, X., Steinfeld, R., Liu, J., Rudolph, C.: An implementation of access-control protocol for IoT home scenario. In: 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), pp. 31–37 (2017). DOI 10.1109/ICIS.2017.7959965