Energy consumption analysis of TSCH-enabled platforms for the Industrial-IoT

Pietro Boccadoro, Michele Barile, Giuseppe Piro, and Luigi Alfredo Grieco
Dep. of Electrical and Information Engineering (DEI), Politecnico di Bari,
v. Orabona 4, 70125, Bari, Italy;
e-mail: {name.surname}@poliba.it.

***Abstract*—Industrial Internet of Things (IIoT) is emerging as a new communication paradigm for a class of low-power wireless networks used in critical applications, such as industrial process monitoring and automation. While these networks are receiving significant attention from standardization bodies, a lightweight protocol stack, built on top of the Time Slotted Channel Hopping (TSCH) technology, is now available. IIoT devices are usually energy-constrained. Thus, the worldwide scientific community is spending a lot of efforts in designing new energy-saving communication protocols for TSCH-enabled networks. Nevertheless, even if the energy consumption is mainly due to radio communications, transmission of unneeded data, protocol overhead, and non-optimized communication patterns, the implementation of the IIoT protocol suite in real platforms may introduce a further and unpleasant waste of energy. The aim of this work is to experimentally measure the energy consumption of TSCH-enabled platforms (specifically TelosB motes) under different system configurations. The two investigated operating systems are OpenWSN and Contiki. Obtained results will demonstrate that OpenWSN always registers a lower energy consumption, thus emerging as a promising protocol implementation for upcoming energy-efficient IIoT systems.**

***Index Terms*—Industrial Internet of Things; TSCH; energy consumption; Contiki; OpenWSN.**

I. Introduction

The revolutionary Internet of Things (IoT) paradigm enabled the interaction among smart objects, pervasively diffused across the Internet [1]. According to the latest report from Gartner[1], in fact, approximately 3.9 billion connected things were in use in 2014 and this value is expected to rise to 25 billion by 2020. The potentials of these connected smart objects can be *already used* for providing advanced services in several domains, like home, enterprise, healthcare, urban, energy, logistics and transportation, education, and entertainment [2]. More recently, the Industrial Internet of Things (IIoT) is emerging as a new class of low-power wireless networks, deployed in critical applications such as industrial process monitoring and automation.

A cornerstone technology for the IIoT is the Time Slotted Channel Hopping (TSCH) [3], already standardized in the IEEE802.15.4e amendment [4] in 2012. In TSCH networks, all nodes are synchronized, and time is cut into timeslots. A timeslot is long enough for a node to send a frame to its neighbor, and for that neighbor to send back a link-layer acknowledgment indicating the successful reception. Moreover, the whole communication is orchestrated by a schedule which indicates, to each node, what to do in each slot: transmit, listen or sleep. On top of the TSCH, the Internet Engineering Task Force (IETF) defined a lightweight protocol stack, properly designed by considering the constrained nature of IIoT devices, as well as the communication requirements (reliability, security, latency, bandwidth, etc.) of typical industrial applications [1]. Today, the protocol suite has already been implemented in many operating systems, such as OpenWSN [5] and Contiki [6], and it is ready to be used in real deployments.

It is extremely important to remark that IIoT devices are usually energy-constrained, since they are powered by batteries or through energy-harvesting [7], [8]. Therefore, the scientific community consider the development of energy-saving communication protocols for the IIoT as one of the hottest research topics. In fact, if from one side energy consumption is mainly due to radio communications, from another side it can be wasted by transmission of unneeded data, protocol overhead, and non-optimized communication patterns, and so on. What however is still unclear is the fact that the implementation of the IETF protocol stack on real platforms may introduce a further and unpleasant waste of energy.

Based on these premises, the present contribution aims at investigating, through real experiments, the energy consumption of both OpenWSN and Contiki firmware implementations for the TelosB platform. To this end, three different system configurations have been taken into account. They are: (i) an isolated node with LEDs off that does not use the radio interface for transmitting/receiving data; (ii) an isolated node with one of the onboard LEDs turned on for debugging purposes, but the radio interface is still not used for transmitting/receiving data, and (iii) a node with one of the LEDs on and with its radio interface in use for transmitting/receiving data. Experimental tests demonstrate that OpenWSN appears as the most energy-efficient operating system in all the evaluated conditions.

The rest of the paper is organized as in the following. Section II describes the protocol stacks conceived for the IIoT, along with its implementation in the two considered operating systems. The experimental setup and the obtained results are illustrated in Section III and Section IV, respectively. Finally, Section V provides the conclusions.

[1]http://www.gartner.com/technology/research/internet-of-things/, latest accessed: 2016-04-12

II. IIoT standards and their implementations

IIoT networks are receiving significant attention from standardization bodies, which conceived communication schemes, network architectures, and algorithms very suitable for low-power and short-range wireless communications. As shown in Figure 1, the resulting protocol suite is composed of:

- IETF CoAP: application protocol which easily translates to HTTP for integration with the web, while meeting specialized requirements such as: multicast support, very low overhead, and simplicity for constrained environments [9].
- IETF RPL: gradient based routing protocol that can ease the formation and the management of multi-hop topologies based on short-range low-power links. It supports multiple roots and is highly flexible thanks to the possibility to optimize the topology based on parametric optimization functions [10];
- IETF 6LoWPAN: adaptation layer to let IPv6 datagram to fit the small payload size (up to 127 bytes in IEEE 802.15.4) by means of advanced header compression techniques [10];
- IETF 6TOP: adaptation layer that enables the integration between higher-layers protocol and the novel IEEE802.15.4e standard through management and data interfaces and organize the transmission of a IPv6 packet over a TSCH protocol [3];
- IEEE 802.15.4e-2012 MAC: powerful MAC layer based on the TSCH protocol, which ensures reliability and energy efficiency in challenging wireless PAN [4];
- IEEE 802.15.4-2011 PHY: low-power physical layer based on the Direct Sequence Spread Spectrum (DSSS) modulation scheme and operating at the 2.4 GHz of the ISM band [11];

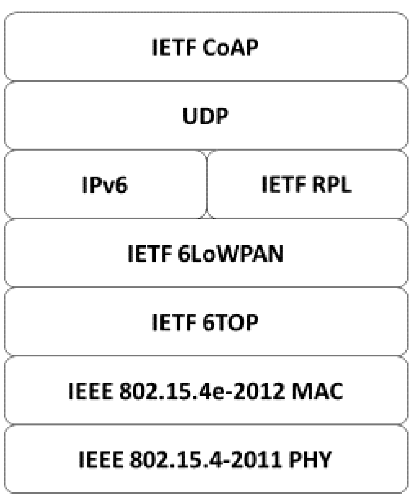

Fig. 1: Protocol suite for the IIoT.

At the time of this writing, there exist two widely accepted firmware implementations for the presented protocol stack. They are OpenWSN [5] and Contiki [6].

A. OpenWSN

OpenWSN is an open-source operating system developed at the Berkeley University, which offers a high compatibility with a huge number of boards and sensors.

1) Memory Footprint: it needs for 30 kB of ROM and takes almost 10 kB of RAM.

2) Software details: it is written in C. Beside the protocol stack reported in Figure 1, OpenWSN also implements applications that use on-board sensors. A dedicated software written in Python, specifically Open Visualizer [12], is used to control the status of motes, quality of links, scheduling of resources, etc.

3) Time slot management: as default, the time slot duration is set to 15 ms. The transmission of the data frame and the corresponding ACK takes at most 4.6 ms and 1 ms, respectively. The rest of the time slot is dedicated to the management of the radio interface, data processing, and security operations (if enabled[2]).

B. Contiki

Contiki is an open-source and community-driven operating system.

1) Memory Footprint: similarly to OpenWSN, it needs for 30 kB of ROM and takes almost 10 kB of RAM.

2) Software details: it is written in C. Differently from OpenWSN, Contiki does not provide a software interface for handling live debugging functionalities. However, these functionalities can be still used during the developing phase of new version of protocols and algorithms, by means of instruments offered with the Cooja simulator.

3) Time slot management: by considering the baseline time slot duration as for OpenWSN (i.e., 15 ms), Contiki has a heavy radio interface management that brings to higher energy consumption (see Section IV for more details).

III. The experimental setup

A. The adopted hardware platform

The hardware platform used in this work is the Telos Rev B board [13], widely referred to as TelosB. It integrates the TI CC2420 radio interface and the MSP430 16-bit microcontroller. The board also has 3 LEDs, 2 push buttons, and sensors measuring temperature, humidity, and light (both total and photosynthetical).

The microcontroller can work in 2 different operative modes: ACTIVE (i.e., it executes some operations) and SLEEP (i.e., no operations are executed). The radio-frequency transceiver, instead, can work in RECEIVE (i.e., the radio interface is turned on for transmitting/receiving data), IDLE (i.e., the radio interface is turned on, awaiting for data), and SLEEP (i.e., the radio interface is turned off) modes.

The TelosB platform is fed by two rechargeable batteries having a resulting dropdown voltage equal to 2.4 V. Moreover, according to the reference datasheet, the current drawn in all the aforelisted operative modes is reported in Table I.

[2]Security in IEEE 802.15.4e-2012 is optional and it is not taken into account in this work.

Operative mode	Current Draw
Microcontroller - active mode	1.8 mA
Microcontroller - sleep mode	5.1μ A
Radio - active mode RX	18.8mA
Radio - active mode TX	17.4mA
Radio - idle mode	21μ A
Radio - sleep mode	1μ A

TABLE I: Current drawn for different operative modes for both board and RF transceiver [13].

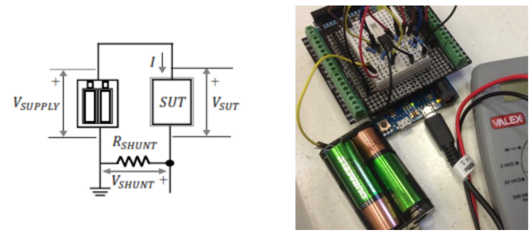

Fig. 2: Schematic and picture of the experimental test.

B. Reference configurations

To properly evaluate the amount of energy consumed by TSCH-enabled platforms, as well as clearly identifying the impact of the protocol stack implementation, we considered three different configurations:

- **Configuration 1**: the test aims at measuring the energy consumption related to the pure operating system. Therefore, the investigated TelosB platform is isolated (i.e., it does not transmit data and it is not synchronized with any another device). Moreover, its radio interface and all the LEDs are turned off.
- **Configuration 2**: the test aims at measuring the energy overhead introduced by a LED turned on for debugging purposes. Similarly to the previous configuration, also in this case the radio interface is turned off.
- **Configuration 3**: the test aims at measuring the overall amount of energy consumed when transmitting data (e.g., a short test message) during a single TSCH slot.

The memory footprint required to implement the three reference configurations is reported in Table II. These values demonstrates that OpenWSN is average 39% (35% for Configuration 1, 35% for Configuration 2, 46.78% for Configuration 3) lighter in RAM than Contiki. As for ROM memory, OpenWSN is average equivalent to Contiki (1.5% more for Configuration 1, 1.5% more for Configuration 2, 3% less for Configuration 3).

C. Measurement setup

To perform the energy consumption analysis of TSCH-enabled networks, it has been chosen to measure the current that flows through a Shunt Resistor, a simple and effective method which does not interfere with the System Under Test (SUT). Figure 2 shows the deployed measurement setup, including its schematic. Values of the current have been revealed with a resistive component connected to a MCP6041-I/P OPAMP [14] and Arduino DUE (a microcontroller-based prototyping board with an 12-bit ADC) [15]. Also, the testbed integrates a resistor of 1 $\Omega \pm 1\%$.

IV. Experimental Results

The current drawn due to the sole operating system (measured with the first reference configuration) has been reported in Figure 3. From results, it is evident that Contiki registers a higher current absorption (specifically, 0.818 mA versus the average value of 0.224 mA measured for OpenWSN). Furthermore, the power consumption is equal to 2.454 mW for Contiki and 0.672 mW for OpenWSN[3]. Thus, OpenWSN guarantees an energy-saving equal to 27.38%.

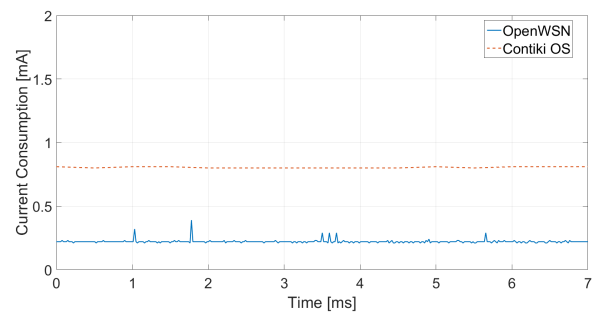

Fig. 3: Current Consumption in Configuration 1.

As expected, in the case one of the LEDs available on the TelosB platform is turned on for debugging purposes (the test refers to the second reference configuration), an additional amount of energy is wasted. From Figure 4, it emerges that the current absorption is equal to 1.76 mA for Contiki and 0.762 mA for OpenWSN, thus resulting to a overall power consumption of 5.28 mW and 2.28 mW, respectively. Now, the energy-saving offered by OpenWSN increases to 43.18%.

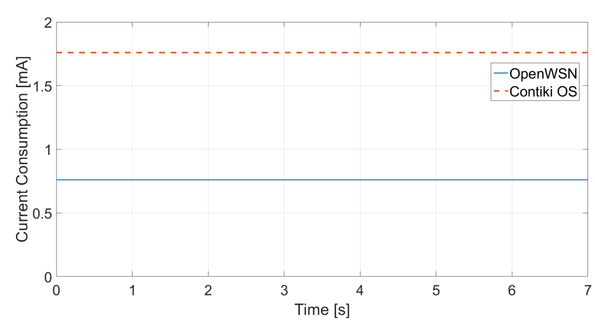

Fig. 4: Current Consumption in Configuration 2.

To provide a further insight, the current drawn registered during the whole TSCH slot for both OpenWSN and Contiki implementations is shown in Figures 5 and 6, respectively.

From results it emerges that:

[3]The power consumption is calculated by multiplying the dropdown voltage and the measured current consumption.

	Tests		
	Configuration 1	**Configuration 2**	**Configuration 3**
ROM footprint for OpenWSN	44332 B	44336 B	44280 B
RAM footprint for OpenWSN	4132 B	4132 B	4266 B
ROM footprint for Contiki	43672 B	43678 B	45612 B
RAM footprint for Contiki	6428 B	6428 B	8016 B

TABLE II: Memory footprints.

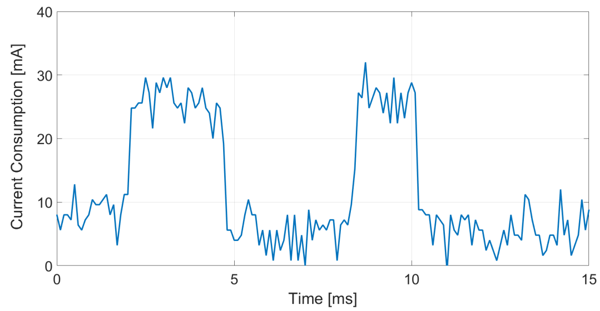

Fig. 5: Current consumption in a slot, registered with the OpenWSN implementation.

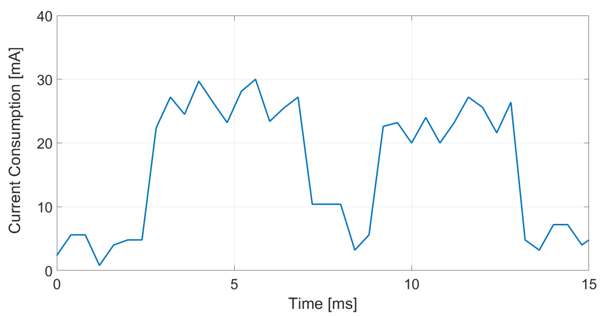

Fig. 6: Current consumption in a slot, registered with the Contiki implementation.

- when the Contiki implementation is used, a data packet is transmitted in 3.5 ms and the process consumes 0.219 mJ[4];
- when the Contiki implementation is used, the corresponding ACK packet is transmitted in 2.5 ms and the process consumes 0.105 mJ;
- the Contiki implementation registers an overall energy consumption during the TSCH slot equal to 0.324 mJ;
- when the OpenWSN implementation is used, a data packet is transmitted in 2.5 ms and the process consumes 0.15 mJ;
- when the OpenWSN implementation is used, the corresponding ACK packet is transmitted in 2 ms and the process consumes 0.111 mJ;
- the OpenWSN implementation registers an overall energy consumption during the TSCH slot equal to 0.261 mJ;

Definitively, OpenWSN emerges as the most energy-efficient operating system. It, in facts, registers an energy-saving equal to 23.67%.

[4]The energy consumption is obtained by multiplying the consumed power and the observation time.

V. Conclusions

This work experimentally evaluates the amount of energy consumed by a commercial platform for the IIoT (i.e., the TelosB mote), when it runs two widely used operating systems, that are OpenWSN and Contiki. Conducted tests considered three different configurations: (i) an isolated node with LEDs off that does not use the radio interface for transmitting/receiving data; (ii) an isolated node with one of the onboard LEDs turned on for debugging purposes, but the radio interface is still not used for transmitting/receiving data, and (iii) a node with one of the LEDs on and with its radio interface in use for transmitting/receiving data. Moreover, obtained results demonstrated that the OpenWSN operating system ensures an energy-saving up to 24%, thus resulting in the most energy-efficient implementation of the IIoT protocol stack. For the future works, we plan to further improve our investigation by deeply characterizing these operating systems in other commercial platforms. Specifically, we will consider their impact on energy consumption, duty cycle, and networking functionalities (like system bootstrap, resilience of communication links to the quality of the wireless transmission), as well as their integration in advanced network architectures based on the emerging information-centric communication paradigms.

VI. Acknowledgment

This work was partially supported by the BONVOYAGE project, which received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 63586, and by the project SymbIoTe, which receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 688156.

References

[1] M. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. Grieco, G. Boggia, and M. Dohler, "Standardized Protocol Stack for the Internet of (Important) Things," *IEEE Surveys & Tutorials.*, vol. 15, no. 3, 2013.

[2] J. Kim, J. Lee, J. Kim, and J. Yun, "M2m service platforms: Survey, issues, and enabling technologies," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 61–76, Jan. 2014.

[3] M. R. Palattella, T. Watteyne, Q. Wang, K. Muraoka, N. Accettura, D. Dujovne, L. A. Grieco, and T. Engel, "On-the-fly bandwidth reservation for 6tisch wireless industrial networks," *IEEE Sensors Journal*, vol. 16, no. 2, pp. 550–560, Jan 2016.

[4] *802.15.4e-2012: 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., Apr. 2012.

[5] Regents of the University of California, "OpenWSN," April 2016. [Online]. Available: https://openwsn.atlassian.net/wiki/pages/viewpage.action?pageId=688187

[6] Contiki Developers Community, "Contiki OS," 2016. [Online]. Available: http://www.contiki-os.org/

[7] H. Jayakumar, A. Raha, Y. Kim, S. Sutar, W. S. Lee, and V. Raghunathan, "Energy-efficient system design for iot devices," in *Proc of IEEE Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan. 2016, pp. 298–301.

[8] N. Kaur and S. K. Sood, "An energy-efficient architecture for the internet of things (iot)," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–10, 2015.

[9] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, *Constrained Application Protocol (CoAP)*, IETF CoRE Working Group, Feb. 2011.

[10] O. Hersent, D. Boswarthick, and O. Elloumi, *The Internet of Things: Key Applications and Protocols*. Wiley, 2012.

[11] IEEE std. 802.15.4, *Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, Standard for Information Technology Std., Jun. 2011.

[12] T. Watteyne, X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wang, S. Glaser, K. Pister, "Openwsn: A standards-based low-power wireless development environment," *Wiley Transactions on Emerging Telecommunications Technologies*, vol. 23, no. 5, pp. 480–493, 2012.

[13] Zolerzia SL, "Zolertia z1 datasheet," 2010.

[14] Microchip Technology Inc., "Mcp6041-i/p datasheet," 2013.

[15] Arduino DUE, "Arduino due datasheet," 2012.