HAL
open science

Improving Low Power Listening (LPL) Mechanism to Save Energy Consumption in WSN

Jessye dos Santos, Guillaume Terrasson, Alvaro Llaria

To cite this version:

Jessye dos Santos, Guillaume Terrasson, Alvaro Llaria. Improving Low Power Listening (LPL) Mechanism to Save Energy Consumption in WSN. SENSORS 2020, Oct 2020, Rotterdam, Netherlands. hal-03020943

HAL Id: hal-03020943

https://hal.science/hal-03020943v1

Submitted on 24 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Improving Low Power Listening (LPL) Mechanism to Save Energy Consumption in WSN

Jessye Dos Santos, Guillaume Terrasson and Alvaro Llaria
Univ. Bordeaux, ESTIA INSTITUTE OF TECHNOLOGY, F-64210 Bidart, France
Email: {j.dossantos,g.terrasson,a.llaria}@estia.fr

***Abstract*—As stated in the literature, Low Power Listening (LPL) duty cycle is one of the most common energy conservation solution for WSN. By using channel check mechanism, the purpose of LPL solutions is to reduce the energy consumption of the listening phase. In this paper, we propose to study the performances and limitations of this kind of solutions. Therefore, we deploy a ContikiMAC LPL on both real and simulated WSN platform to demonstrate the impact of LPL on the energy consumptions of the node radio and microcontroller but also on the application Quality of Service. Based on the obtained results, shortcomings of LPL solutions are highlighted and potential improvements are discussed such as the use of multi-parameter dynamic duty cycle.**

I. Introduction

Wireless Sensor Networks (WSN) consist of hundreds or thousands of small devices called nodes. Nodes are in general composed of a microcontroller (MCU), memories and a wireless transceiver (RF) [1]. They also embed heterogeneous sensors (temperature, humidity, accelerometers...) to monitor their environments. Thanks to standards as 6LoWPAN [2], WSN nodes become smart and can be part of Internet of Things (IoT) paradigm.

Therefore, deployed at different areas, these tiny devices allow various applications as smart agriculture [3]–[5], Industry 4.0 [6]–[8] or smart campus/smart home [9]–[11]. All these applications need robust and reliable technologies for collecting and sending data. Nevertheless, sensor nodes are currently powered with batteries which leads to limit WSN lifetime [12]. So, it is essential to propose and develop energy optimization solutions aimed at reducing node energy consumption without downgrading the network Quality of Service (QoS) [13].

As exposed in the literature, these solutions can be implemented at hardware or software levels [14]. For example, in [15] and [16], the developed nodes embed hardware solutions which manage parameters such as MCU frequency, supply voltage or current of the components in order to save energy. As pointed out in [14], this kind of hardware solutions cost the creation or addition of components and cannot be simply deployed on existing nodes. On the other hand, software solutions, can be deployed on each OSI layers [17] [18]. As an example, duty cycle methods, located on MAC layer, adapt the use of the hardware states (sleep, idle, ON...) with the application needs. Combined with the progress made to propose hardware running at lower power states [12], the use of duty cycle seems to be the most efficient solution to save energy in WSN [19].

Consequently, several approaches of duty cycle have been proposed in the literature [20]. Because synchronization is energy intensive and difficult to deploy on WSN, the most adapted duty cycle techniques for WSN are asynchronous. In this context, Low Power Listening (LPL) solution is one of the most popular energy saving solutions [21] embedded as default solutions in the two common open source OS dedicated to WSN [22], Contiki OS [23] and Tiny OS [24]. In LPL, the node performs periodical wake-up to check the channel to detect possible communications from neighbor nodes. If a signal is detected, node stays awake to receive data and to route it if necessary. Otherwise, the node comes back to sleep state to save energy until the next channel check. By this mechanism, nodes can improve energy consumption by keeping network reliability. When a node wants to send, it just performs, before sending data, Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) protocol with Clear Channel Assessment (CCA) to be sure the media is free.

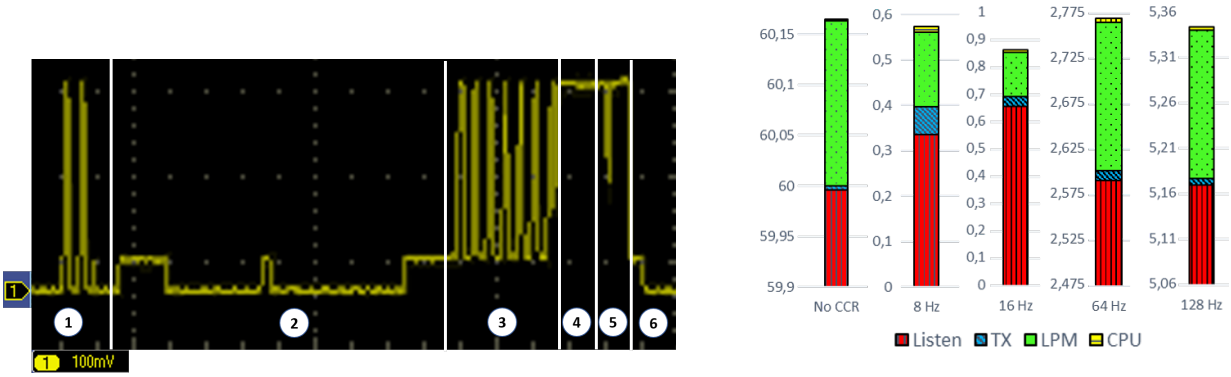
In this paper, performances of LPL based duty cycle solution are analyzed through the ContikiMAC [25] example. Wake-up interval called Channel Check Rate (CCR) determines the sleep/wake-up scheduling. It is predefined before WSN deployment and so it is static. Therefore, the impact of CCR on energy consumption and on Packet Rate Reception (PRR) and packet retransmissions are studied. Using the obtained results, shortcomings linked to CCR configuration are highlighted and solutions to improve LPL process are proposed and discussed.

The rest of this paper is organized as follows. In section II performance results obtained with our deployed platforms are described. Section III discusses countermeasures to improve LPL and the remaining issues. Then, we conclude and present future works.

II. Performances analysis

In ContikiMAC, apart from application time, the duration the node can sleep to save energy is defined thanks to CCR. Different values of CCR can be configured. This CCR configuration must be made before node deployment and so it is static.

When nodes are in sleeping mode, no task is performed. Therefore, to study the impact of CCR values on WSN applications, important criterion to analyze are not only the



(a) Profile obtained with real testbed.

(b) Energy consumption from collect view tool.

Fig. 1. Energy consumption results.

energy consumption but also network performances such as PRR and the number of retransmitted application data. PRR represents the number of packets received compared to the number of packets sent. A 100% of PRR indicates application runs without disturbances. If no retransmission is necessary, more energy is saved. So, we study the impact of CCR values on these two parameters to quantify the QoS.

TABLE I
DESCRIPTION OF THE EXPERIMENTAL ENVIRONMENTS.

	Testbed	Simulation
Node Hardware	XM1000 [26]	Z1 [27]
Energy Monitoring System	1Ω shunt resistor 5% AOP AD620ANZ [28] Osc. Tektronic MSO 2014 [29]	Cooja Collect View [30]
QoS Monitoring System	Sniffer + Wireshark [31]	Python software

An experimental testbed and a Cooja [32] simulation setup are proposed to analyze these performances. Table I summarizes the hardware used for each setup, the monitoring systems for energy and network performance evaluations. On testbed, we profile the global energy consumption of a XM1000 node. In this study, simulation platform allows us to have a more fine-grained analysis and to separate consumptions of MCU with RF (Radio ON, Receiver state (RX) and Transmitter state (TX)) but sensors and buses cannot be accessed. These missing components do not impact this study because CCR has no effect on them.

During experiments, after joining the network, End Device (ED), implementing ContikiOS 3.0, sends periodically sensors data (testbed) or a counter value (simulation) to a sink node. ContikiMAC wake-up is used as duty cycle solution on ED. Sink node keeps radio 100% ON.

A. Energy consumption investigation

Fig. 1a shows the energy profile obtained during sending data (application) phase with our experimental testbed for a node implementing ContikiMAC. As demonstrated, the ContikiMAC protocol performs two periodical consecutive CCA to check the media ①. Next application starts and, at the end of ②, the MCU is waked up. After that, before sending data, the node performs 6 CCA to be sure the media is free ③. These 6 CCA are due to CSMA-CA protocol which is implemented as part of ContikiMAC protocol. Then the node sends its data ④ and switches on RX state to wait an acknowledgment ⑤. Finally, the MCU goes back to sleep mode ⑥.

As illustrated by Fig. 1b, when ContikiMAC is enabled, the total energy can be divided approximately by 104 for CCR at 8Hz compared to No CCR. RF represents at least 70% of total energy consumption and until 99.7% when No CCR is used. RF spends the most energy in listening phase (① & ⑥ of Fig. 1a). In case of No CCR, because MCU and RF are always ON, listen phase is 59.9mW against 0.335mW with CCR at 8Hz.

TABLE II
DUTY CYCLE VALUES FOR 1S APPLICATION PERIOD.

CCR (Hz)	RF ON %	TX %	RX %
8	1.45	0.29	0.04
16	1.97	0.29	0.04
64	5.42	0.29	0.04
128	10.13	0.29	0.04
No CCR	100	0.29	0.04

Using Cooja setup, the percentage of time the radio spends in RX, TX or ON is shown in Table II for an application sending data every second with different values of CCR. The RX percentage stays equal to 0.04% (respectively 0.29% for TX) even when CCR changes. For the same application period, duty cycle (RF ON) increases with CCR from 1% to 10% in average. This phenomenon is explained by the listen phase which is multiplied by 7 in average when CCR changes from 8Hz to 128Hz. The node runs more often CCA to check the network which costs energy. Thus, a higher CCR implies a higher energy consumption.

At the origin, these different values of CCR allow to adapt the check frequency with the application needs and the network conditions (topologies, traffic, possible network congestion...) to offer correct QoS. Nevertheless, the obtained results demonstrate that CCR value has to be defined cautiously in the design phase because it could have a major impact on the energy consumption.

B. Network performances analysis

QoS values observed for the testbed are similar to them of simulation but, because network disruptions can deteriorate PRR and retransmissions, we use simulations to deploy repeatable WSN with no losing media to analyze only the impact of CCR values on the QoS.

First, using the scenario of section II-A, for all the values of CCR, PRR is equal to 100% and the retransmission is none. So, a CCR of 128Hz does not improve network performances in case of 1s application period and costs more energy consumption than 8Hz.

After that, the performances of WSN with a CCR defined lowly to save more energy and an application with high periodicity are studied. In this case, the CCR is configured at 4Hz and the application period is chosen below CCR period at 0.2s.

TABLE III
NETWORK PERFORMANCES.

Nb of nodes	Exp.	ED	R	Sink	PRR %	ReTX %
2	1	CCR		NoCCR	100	20
	2	CCR		CCR	99.6	95.4
3	3	CCR	CCR	NoCCR	3.6	94.5

Table III shows the PRR and the number of retransmitted data and routing frames (ReTX) for three kind of experiment. ”CCR” entry indicates node with CCR at 4Hz, ”NoCCR” entry is for node always ON and black grid is when node is not deployed on the experiment.

In Exp. 1, as sink node has no CCR, the PRR is 100%. We can observe 20% of packet retransmitted. This is due to the high frequency of the application. In Exp. 2, when sink node performs CCR too, the retransmission becomes 95.4%. This implies that the ED has no time to perform duty cycle between two transmissions. Nevertheless, the PRR stays high at 99.6%. If topology change as in Exp. 3 and a router R is placed between ED and a sink node always ON, 94.5% of frames are retransmitted for a PRR of 3.6%.

In conclusion, for network performances, when application period is high (1s), additional CCA linked with a higher CCR could be unnecessary in relation to the needs of protocols and application, costing more energy consumption without improving network performances. On the other hand, when CCR is defined inaccurately by choosing above period application to save energy or if network topology changes as in dynamic network, the static hard coded CCR, as demonstrated, is not adapted and the QoS performances can be disrupted. A lot of retransmission happened costing energy and disrupting duty cycle mechanism.

III. HOW TO IMPROVE ENERGY MANAGEMENT?

LPL solution is claimed as the best software solution in literature to improve energy consumption in WSN. Nevertheless, through the results obtained in section II-A, we shown radio energy consumption is multiplied by 7 for a CCR at 128Hz compared to a CCR at 8Hz. Moreover, network performances can be disrupted if CCR period is not adapted to network conditions, as shown in section II-B where PRR falls at 3.6% with 94.5% of retransmitted frames. So, these results demonstrate that static hard coded duty cycle is difficult to configure.

To limit the node energy consumption while keeping network performances suitable, one solution consists to update the value of the duty cycle in run time [21]. This dynamic solution can be based on application needs [33], traffic conditions [34] or incoming/outgoing packet rate [35], or considering the battery capacity level [36]–[38]. Nevertheless, all the previous LPL dynamic solutions use only one unique parameter to update their duty cycle.

Therefore, to preserve more energy, one solution could be to use a multi-parameter dynamic duty cycle. With the monitoring of parameters coming from different sources (sensors, application, hardware, batteries...) and a decision algorithm, the duty cycle can be dynamically managed on run time. This solution allows to take into account more phenomenon linked with the several dynamic changes happening in WSN to adapt the sleep period and so, to save more energy.

Nevertheless, multi-parameter dynamic duty cycle needs to monitor all the previously mentioned parameters. It requires hardware and software tools to collect and provide these data to the decision algorithm. Secondly, this duty cycle updating must be less energy costly than the use of a static one. So, the cost of this new process must be evaluated, and the policy must be determined. Finally, updated duty cycle has to be computed autonomously by the nodes to save more energy. This non-unified duty cycle value can cause missing neighbor communications, retransmissions needs and poor network performances. These issues stay open.

IV. CONCLUSION AND FUTURE WORK

In this article, we shown that when LPL solution is deployed to reduce the listening phase, the static hard coded value of the wake-up period performs unnecessary channel checks and so costs energy consumptions. Moreover, a bad configuration can cause poor network performances. To counter these issues, we have discussed the necessity to propose dynamic duty cycle solutions. Our future work will be dedicated to deployed multi-parameters dynamic duty cycle in Contiki OS. Moreover, to enable our solution to run, issues linked with non-unified duty cycle should be resolved. Work will also be made on development of tools for monitoring parameters useful for the dynamic update of the duty cycle. Monitoring and decision costs will be evaluated to choose the best policy.

REFERENCES

[1] M. Johnson, M. Healy, P. Van de Ven, M. J. Hayes, J. Nelson, T. Newe, and E. Lewis, "A comparative review of wireless sensor network mote technologies," in SENSORS, 2009 IEEE. IEEE, 2009, pp. 1439–1442.

[2] J. Hui, D. Culler, and S. Chakrabarti, "6lowpan: Incorporating ieee 802.15. 4 into the ip architecture," IPSO Alliance White Paper, vol. 3, 2009.

[3] S. Heble, A. Kumar, K. V. D. Prasad, S. Samirana, P. Rajalakshmi, and U. B. Desai, "A low power iot network for smart agriculture," in 2018 IEEE 4th World Forum on Internet of Things (WF-IoT). IEEE, 2018, pp. 609–614.

[4] A. Triantafyllou, P. Sarigiannidis, and S. Bibi, "Precision agriculture: A remote sensing monitoring system architecture," Information, vol. 10, no. 11, p. 348, 2019.

[5] N. Sales, O. Remédios, and A. Arsenio, "Wireless sensor and actuator system for smart irrigation on the cloud," in 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). IEEE, 2015, pp. 693–698.

[6] S. Raza, M. Faheem, and M. Guenes, "Industrial wireless sensor and actuator networks in industry 4.0: Exploring requirements, protocols, and challenges a mac survey," International Journal of Communication Systems, vol. 32, no. 15, p. e4074, 2019.

[7] B. Prabhu, E. Gajendran, and N. Balakumar, "Smart oil field management using wireless communication techniques," Smart Oil Field Management Using Wireless Communication Techniques (January 8, 2017). IJIEST, 2016.

[8] T. Watteyne, V. Handziski, X. Vilajosana, S. Duquennoy, O. Hahm, E. Baccelli, and A. Wolisz, "Industrial wireless ip-based cyber–physical systems," Proceedings of the IEEE, vol. 104, no. 5, pp. 1025–1038, 2016.

[9] T. G. Stavropoulos, G. Koutitas, D. Vrakas, E. Kontopoulos, and I. Vlahavas, "A smart university platform for building energy monitoring and savings," Journal of Ambient Intelligence and Smart Environments, vol. 8, no. 3, pp. 301–323, 2016.

[10] A. Elsts, G. Oikonomou, X. Fafoutis, and R. Piechocki, "Internet of things for smart homes: Lessons learned from the sphere case study," in 2017 Global Internet of Things Summit (GIoTS). IEEE, 2017, pp. 1–6.

[11] O. M. Butt, M. Zulqarnain, T. M. Butt, and A. S. Malik, "Development of smart home automation system by using controlled network," JOURNAL OF FACULTY OF ENGINEERING & TECHNOLOGY, vol. 23, no. 2, pp. 69–76, 2016.

[12] F. Karray, M. W. Jmal, A. Garcia-Ortiz, M. Abid, and A. M. Obeid, "A comprehensive survey on wireless sensor node hardware platforms," Computer Networks, vol. 144, pp. 89–110, 2018.

[13] A. A. Babayo, M. H. Anisi, and I. Ali, "A review on energy management schemes in energy harvesting wireless sensor networks," Renewable and Sustainable Energy Reviews, vol. 76, pp. 1176–1184, 2017.

[14] A. Musaddiq, Y. B. Zikria, O. Hahm, H. Yu, A. K. Bashir, and S. W. Kim, "A survey on resource management in iot operating systems," IEEE Access, vol. 6, pp. 8459–8482, 2018.

[15] A. Silva, M. Liu, and M. Moghaddam, "Power-management techniques for wireless sensor networks and similar low-power communication devices based on nonrechargeable batteries," Journal of Computer Networks and Communications, vol. 2012, 2012.

[16] V. S. Patil, Y. B. Mane, and S. Deshpande, "Fpga based power saving technique for sensor node in wireless sensor network (wsn)," in Computational Intelligence in Sensor Networks. Springer, 2019, pp. 385–404.

[17] H. Patel and V. Shah, "A review on energy consumption and conservation techniques for sensor node in wsn," in 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES). IEEE, 2016, pp. 594–599.

[18] H. M. Jawad, R. Nordin, S. K. Gharghan, A. M. Jawad, and M. Ismail, "Energy-efficient wireless sensor networks for precision agriculture: A review," Sensors, vol. 17, no. 8, p. 1781, 2017.

[19] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: A top-down survey," Computer Networks, vol. 67, pp. 104–122, 2014.

[20] K. A. Memon, M. A. Memon, M. M. Shaikh, B. Das, K. M. Zuhaib, I. A. Koondhar, and N. U. A. Memon, "Optimal transmit power for channel access based wsn mac protocols," Int. J. Comput. Sci. Netw. Secur, vol. 18, pp. 51–60, 2018.

[21] T. Dinh, Y. Kim, T. Gu, and A. V. Vasilakos, "An adaptive low-power listening protocol for wireless sensor networks in noisy environments," IEEE systems journal, vol. 12, no. 3, pp. 2162–2173, 2017.

[22] Z. Wang, W. Li, and H. Dong, "Review on open source operating systems for internet of things," in Journal of Physics: Conference Series, vol. 887, no. 1. IOP Publishing, 2017, p. 012044.

[23] M.-P. Uwase, M. Bezunartea, J. Tiberghien, J.-M. Dricot, and K. Steenhaut, "Experimental comparison of radio duty cycling protocols for wireless sensor networks," IEEE sensors journal, vol. 17, no. 19, pp. 6474–6482, 2017.

[24] M. Amjad, M. Sharif, M. K. Afzal, and S. W. Kim, "Tinyos-new trends, comparative views, and supported sensing applications: A review," IEEE Sensors Journal, vol. 16, no. 9, pp. 2865–2889, 2016.

[25] A. Dunkels, "The contikimac radio duty cycling protocol," 2011.

[26] https://www.advanticsys.com/shop/asxm1000-p-24.html, online; accessed 13 June 2020.

[27] http://zolertia.sourceforge.net/wiki/index.php/Z1, online; accessed 13 June 2020.

[28] https://www.analog.com/media/en/technical-documentation/data-sheets/AD620.pdf, online; accessed 13 June 2020.

[29] https://www.tek.com/datasheet/mso2000-dpo2000-series, online; accessed 13 June 2020.

[30] https://anrg.usc.edu/contiki/index.php/Collect_View, online; accessed 13 June 2020.

[31] https://www.wireshark.org/, online; accessed 13 June 2020.

[32] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in Proceedings. 2006 31st IEEE Conference on Local Computer Networks. IEEE, 2006, pp. 641–648.

[33] R. Yadav and A. Daniel, "Performance and analysis of mac protocols based on application," in AIP Conference Proceedings, vol. 1952, no. 1. AIP Publishing LLC, 2018, p. 020076.

[34] S. Siddiqui, S. Ghani, and A. A. Khan, "Adp-mac: An adaptive and dynamic polling-based mac protocol for wireless sensor networks," IEEE Sensors Journal, vol. 18, no. 2, pp. 860–874, 2017.

[35] E. D. Ayele, J. Wen, Z. Ansar, and W. Dargie, "Adaptive sleep-time management model for wsns," in 2015 24th International Conference on Computer Communication and Networks (ICCCN). IEEE, 2015, pp. 1–7.

[36] G. Segura and C. B. Margi, "Duty cycle based energy management tool for wireless sensor networks," XXXIV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais, 2016.

[37] H. Ayadi, A. Zouinkhi, T. Val, A. Van den Bossche, and M. N. Abdelkrim, "Network lifetime management in wireless sensor networks," IEEE Sensors Journal, vol. 18, no. 15, pp. 6438–6445, 2018.

[38] S. Galmés and S. Escolar, "Analytical model for the duty cycle in solar-based eh-wsn for environmental monitoring," Sensors, vol. 18, no. 8, p. 2499, 2018.