Adding value to WSN simulation using the IoT-LAB experimental platform

Georgios Z. Papadopoulos, Julien Beaudaux, Antoine Gallais, Thomas Noël and Guillaume Schreiner

ICube laboratory, University of Strasbourg / CNRS, France
E-mail: {gpapadopoulos, beaudaux, gallais, noel, schreiner}@unistra.fr

***Abstract*—Validation of protocols and mechanisms is an essential step to the development of object networks in critical domains. Most papers still provide evaluation either obtained through theoretical analysis or simulations campaigns. Yet, simulators and formal models fail to precisely reproduce the unique specificities of the deployment environments those networks have to evolve in. Also, by putting no limits to code complexity and execution, those tools prevent users to apprehend the actual limits of WSN nodes and to propose realistic communication protocols and applications. In this paper, we highlight to what extent the addition of experimentations can significantly improve the value of performance evaluation campaigns. Along with the recent tendency to have algorithmic and protocol proposals facing real environments, it is questionable whether the so obtained results should be considered as scientific or empirical ones. In the former case, reproducibility, stability over time, topology management to cite a few, are a must have for testbeds and real deployments that are used. In the latter, the results should be viewed as a proof-of-concept only, far from independent of the used hardware and encountered conditions at the experimentation time but still critical from the development cycle standpoint. Through some experiments over the IoT-LAB testbed, we aim at demonstrating to what extent some of the simulation setup and conditions from reality could be emulated. We also provide insight on how to obtain the best out of it in a quick and efficient manner. We show that such testbeds would satisfy many expectations (e.g. scientific tool and proof-of-concept validator), thus minding and bridging some of the gaps between theory and practice in WSN. To this end, we here give an overview of available simulation tools, and guidelines on how to transpose simulation setups to the open large-scale IoT-LAB platform.**

I. INTRODUCTION

The last decade saw the emergence of smart objects networks, with the aim of bringing life to any everyday life object by making it connected. By doing so, a new way to interact with our environment emerged, bringing ubiquity one step further. This paradigm already covers a wide range of applications, from home automation [1] to patient monitoring [2], and imply an infinite deployment cases and conditions. We can observe that there is a tendency where more and more researchers rely on testbeds or real deployments in order to evaluate the performances and behavior of their contributions. Yet, the various environments in which those networks evolve are often too complex to be modeled accurately, and the available analytical tools (e.g. simulators, theoretical models) fail to render the exact environmental conditions [3]. For instance, the radio environment dynamics (e.g. link stability and symmetry over time [4], impact of weather on communications [5]) are mostly unpredictable and can bear a great influence on the deployment success or failure [6].

The reason to this situation is that real experimentations are considered as difficult and complex to set up, and require expertise in either electronics or embedded systems. In the end, many researchers consider that empirically analyzing the performances of their contributions would require far too much time, when compared to the potential gains they can expect. In this paper, we aim at demonstrating that these assumptions are groundless, and that experimental analysis can be easily conducted and add significant values to simulation results, as it has recently been done for formal analysis [7].

We will point out what thorough empirical campaign can bring to the evaluation and analysis. We will characterize the advantages that experiments have over simulations, and identify what goals can be drawn in order to make them complementary. Then, we will provide guidelines to translate simulation campaign to successful experimental deployments. At this stage, we will remind what parameters to take into account and how exactly to set up an experiment to remain in accordance with both the simulations (in order to be able to compare the results) and the final application (to demonstrate the realism of the contribution for future real deployments).

To do so, we will present IoT-LAB, a very large-scale open experimental platform, and demonstrate that most of real deployment conditions can be reproduced accurately on it. We will also explores some of the possibilities offered by this platform. Finally, we will demonstrate, through a series of experiments, how to perform meaningful evaluation in real conditions with only limited time investment.

II. MOTIVATION

There is a trend where research papers on algorithms and protocols evaluate their contributions through simulation. The main reason to this phenomenon is that building a testbed is a very costly procedure and even more, running real experiments is time consuming, complex and difficult. Even if network simulators are the reference validation tool for researchers of our field, their reliability was only marginally investigated. Many of simulators provide abstraction interfaces (e.g. simpler programming language with less low-level functions and electronic references) to hide the complexity of embedded systems programming, thus, allow users to isolate different factors by tuning configurable parameters. They are also designed to facilitate the obtention of all the required results and ensure efficient retrieval of exploitable information. For example, logging operations (e.g. printing of data, calculations) have no impact on the node functioning (e.g. energy consumption, processing time). Moreover, to each information can be associated a precise timestamp (all nodes sharing the same clock). These features enable very detailed and accurate evaluations of any phenomenon (e.g. transmission one-hop delay), and allow for correlation of the data (even those originated from distinct nodes).

Yet, to enable such features, simulations have to hide the limitations and constraints imposed by actual components. For instance, most simulators provide unlimited memory and computation resources and do not take into account the probability of node failure. In this context it is often tricky for users to evaluate the realism and complexity of their contributions. Many researchers develop extensively complex and thus unrealistic mechanisms which might display very promising results in simulation.

Simulators such as Tossim [8] or Cooja [9] were designed to bridge the gap between simulation and experimentation, by remaining as close as possible to programming conditions of real embedded systems. To do so, they rely on the same code for both simulation and experimental campaigns, and thus allow users to really apprehend the specificities of this field as early as at the simulation setup. However, many parameters relative to the radio environment (e.g. link properties and evolution over time, interferences) and node behavior (e.g. failure) cannot be modeled accurately. Thus, even though the code complexity remains identical, the deployment conditions and ensuing effects can only be apprehended and experienced through experimentations.

Simulation procedure is an important phase and a provisioning step during the application or protocol design. Even if the simulations may present good results of the mathematical model. Experiences through past real deployments show that it is not recommended to continue directly with real deployments since they may bear with number of unexpected failures such as node crashing or network disconnection [10], [11]. Hence, they should first test their model on testbeds in order to reflect the reality that their solution would face during real deployments. A testbed is a platform for experimentation of large development projects which allows for rigorous, transparent, and replicable testing of new models and technologies. It appears strategic and crucial to offer to researchers and developers accurate and robust physical large scale testbeds to benchmark, tune, and optimize their solutions. In fact, there is a strong demand among researchers to access the testbed resources required to conduct their experiments.

In theory, people should first test their solutions in simulators (and modeled in maths even before) before proceeding to testbed experiments. An experiment in real testbed platform is a time consuming operation due to its complexity, and even more due to the varying performance (e.g. differences in energy consumption) due to hardware or weather dependencies. When proceeding through testbed steps, people should not expect the same guarantees and validations as the ones provided by the simulators. These drawbacks of testbeds should not fear the people. In fact they should be seen as progressing steps towards real deployments. Researchers obtain meaningful results (e.g. using real radio interfaces, environments with instabilities or with potential defective hardwares) through running real experiments that allow them to get a flavor of the performance of their solution.

However, certain stability must be preserved throughout the development procedure (i.e. modeling, simulations and testbed) of an application or a protocol. Stability is a relevant concept among these steps. In particular, in modeling the algorithm is well-founded through the mathematical formula. In simulation, a researcher has full control of the system and conditions (e.g. environment and hardware). Thus, this latter is established accurately in accordance with the mathematical model in a simulator. Finally, the testbed being an uncontrolled and heterogenous platform, the experiment should be stable in terms of topology, communication links, over time (one experiment run on friday should not differ from another run on monday). In case there are variations in testbed experiments, they should be only due to the used hardware in order to reflect target objects complexity and heterogeneity. Hence, researchers should continue with real deployments as soon as they reach stability of their testbed solution in both three steps.

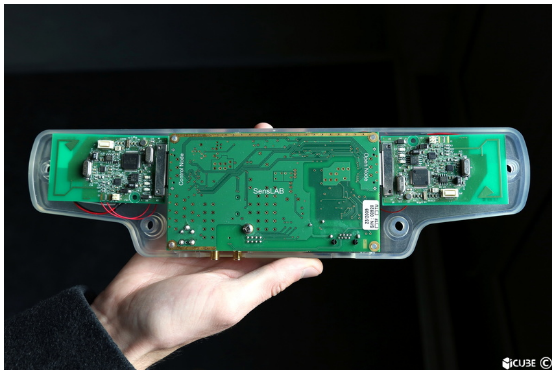
Testbeds and simulators are two important and complementary design and validation tools; an ideal development process should start from the theoretical analysis of the protocol providing bounds and indication of its performance, verified and refined by simulations and finally confirmed in a testbed. Hence, once this procedure is successfully done, researchers could push their solution to engineers.

III. IoT-LAB's Infrastructure

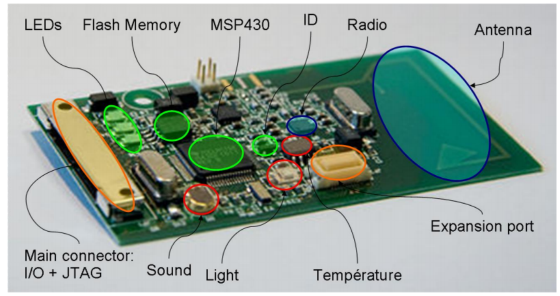
In this section, we introduce IoT-LAB, a very large scale and an open access multi-user testbed. We give a brief description of the hardware (e.g. motes and radio), platform, and testbeds of IoT-LAB.

IoT-LAB platform is composed of 1024 nodes equally distributed over four sites (Strasbourg, Lille, Grenoble and Rennes) interconnected by Internet. Each site has its own characteristics such as fixed or mobile, indoor or outdoor nodes with different radio chips and finally, either isolated or realistic environment, thus, people can expect different behaviors from each site. IoT-LAB provides all necessary tools and experimental facilities for researchers to perform real experiments in large scale platform in order to evaluate and analyze their solutions. Hence, with IoT-LAB testbed, it is possible to translate most of real deployment situation in it.

The testbed platform provides a lot of tools (e.g. RSSI mapping, link quality analyzer) to ensure that deployment conditions are fulfilled. IoT-LAB designed their own board, named WSN430, in order to meet with the requirements in terms of energy monitoring, reproducibility. The nodes of IoT-LAB are running on top of wide spread low power MSP430 micro-controller (see Figure 1(b)) suitable for real WSN deployments [4], [12]. Each node is equipped either with a Texas Instruments CC1101 or CC2420 radio chipset. Hence, the nodes are compatible both with the IEEE 802.15.4 standard



(a) Detailed view of an IoT-LAB node and its gateway



(b) WSN430 board

Fig. 1: The IoT-LAB nodes are protected with a specific box designed on purpose.

and with open Medium Access Control (MAC) protocols. As a result it can communicate with number of devices sharing the same physical layer, including devices from other vendors [13].

Figure 1(a) gives an overview of a IoT-LAB node. In particular, the IoT-LAB hardware components consist in the following three main components: *a)* The **open wireless sensor** node (able to retrieve information for light, sound and temperature) that is dedicated to researchers for their experimentation. Users could flash and run any operating system in the nodes. For their convenience TinyOS [14] and Contiki OS [15] are more suitable since both are using exactly the same code in simulation (i.e. Cooja [9] or Tossim [8]) and in experimentation; *b)* The **full IoT-LAB node** that includes a gateway and a closed wireless node beside the open node. The gateway provides connection to the global infrastructure of the IoT-LAB in order to flash, control and monitor the open node. The closed node is the same as the open one and it is used to interact, passively or actively with the open node (e.g. creating an environment with interference); *c)* The **global networking backbone** that provides power and connectivity to all IoT-LAB nodes.

IoT-LAB lowers the entry cost to real experimentation, often considered as a difficult, complex and heavyweight activity, with no extra management burden, accelerating proof-of-concept evaluation and competitiveness. In the following section, we present number of experimental results in order to describe the behavior of the platform, and to show how to retrieve meaningful results through a simple code.

IV. Experimental Evaluation

Setting up a complete WSN deployment is a very complex task [16]. In this section we present our thorough study on the IoT-LAB deployment. In our experiments, we used the platform installed in Strasbourg, France. This testbed is structured as a 3D grid with 10 lines and 8 columns, distributed in three layers of 80 nodes each, thus, in total 240 fixed nodes. Moreover there are 4 mobile nodes placed on a train (see Figure 2).

Fig. 2: FIT - site of Strasbourg, France

A. Application, routing and MAC details

In our set of experiments, we chose the 80 nodes of the middle layer to work on our project. All 80 nodes are randomly selected as data sources. All nodes implement a time-driven application model in order to avoid the congestion, and broadcast in Constant Bit Rate (CBR) mode one packet (i.e. 100 packets in total) every 60 seconds in the network. We chose a 10 byte data size, which corresponds to the general information used by monitoring applications (e.g. node ID, packet sequence, sensed value) [16]. We run number of experiments with various transmission power (e.g. ranging from $-10, -30dBm$) in order to study the impact of the radio model in transmission range and in link quality. The experiments lasted for 150 minutes while 79000 events occurred in each of them.

In this campaign, we aimed at removing assumptions one at a time. We tried to keep our application as simple as possible and to show how to get meaningful results from simple code. Hence, we run our evaluation within an idealistic scenario where no routing protocol is running in the network which would generate a virtual routing tree rooted at the sink. As a result, this scenario will allow us to focus on IoT-LAB platform performances by affecting the results as little as possible. We picked the well known X-MAC [17] protocol among various preamble-sampling protocols to analyze the IoT-LAB testbed. We kept the standard configuration for X-

MAC (i.e. preamble duration of $500ms$). For convenience in terms of complexity, we implemented our project on top of the Contiki OS [15] since it uses the same code with Cooja simulator [9]. A detailed overview of our experimental parameters is presented in Table I.

TABLE I: Experimental Setup

Experimental parameters	**Value**
Deployed nodes	80 fixed sensors (all being potential sources)
Testbed organization	$(10m \times 8m \times 3m)$ regular grid
Node spacement	One meter
Duration	$150minutes$
Application model	Time-driven: CBR $1pkt/60s$
Payload size	$10Bytes$ (+6 byte MAC header)
Events	7900 events triggering broadcast transmissions
MAC model	X-MAC [17]: $LPL = 500$
Hardware parameters	**Value**
Antenna model	Omnidirectional CC1101 interface
Modulation model	76.8 kBauds GFSK
Transmission power	$-10, -15, -20, -25, -30dBm$
Battery	$880mAh$, $3.7V$

B. Experimentation results and analysis

In this subsection, we present the performance evaluation of IoT-LAB testbed. We analyze the platform in terms of the density of neighbors, the stability of the testbed, the quality of the links and the energy consumption.

a) Density of Neighbors: in Figure 3, the density of neighbors per transmission power is depicted. In our experiment setup, we decided to keep high-quality symmetrical links only (i.e. over than 90% of successful receptions in both ways), in order to provide a fair analysis. As we could expect, the higher the transmission power, the more neighbors. It is worth to point out that the pairs -30 and $-25dBm$, and -15 $-10dBm$ have nearly the same number of neighbors in average. In order to understand this behavior, we calculated the transmission range by using the Euclidian distance equation. The results provided in Table II show us that there is straight forward impact of transmission power with average distance and consequently with average number of neighbors (as expected with our regular topology). It shows a direct translation between transmission power (hardware parameter) and distance. Therefore, the density can be anticipated, as in simulations.

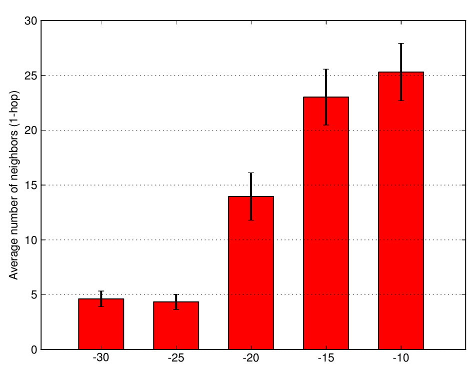


Fig. 3: Number of 1-hop neighbors per transmission power.

TABLE II: Transmission range

Transmission Power	**Range (m)**	**Confidence Interval**
-10 dBm	4.45	0.23
-15 dBm	4.21	0.26
-20 dBm	3.54	0.24
-25 dBm	2.12	0.13
-30 dBm	2.16	0.15

b) Link quality analysis: there are some specific problems that neither occur during simulation nor with only few nodes. Among those, links with low quality induce packet loss and therefore may introduce a bias in the measurements. Hence, in realistic deployments researchers should take into account that most communication links are unstable and unreliable, so that links are often unidirectional. Still, most of routing and MAC protocols require bidirectional links so that two nodes can exchange information (e.g. acknowledgements, data), and it is critical to evaluate wether or not this assumption can be preserved in testbeds. Therefore, we aimed to provide a thorough analysis of the link quality of the IoT-LAB testbed.

Several criterions are expandable to evaluate the quality of a link. Among those, we focus on (Packet Reception Rate) PRR performance. We measured the PRR for every link among the nodes and we computed the link bidirectionality. Hence, we focused on scenario with transmission power of $-20dBm$, and evaluated the stability of links over time. In particular, we repeated this experiment three times every 12 hours, once on Wednesday evening and twice on Thursday (morning and evening). Then, we analyzed the links in order to study their stability in time. In Figure 4, the nodes are depicted in a circle for visualization purpose. The ten nodes are a subset of the testbed nodes that cover the entire second layer of the Strasbourg platform. We can observe that 3 connected components remain stable over time (links 1-3, 2-8 and 4-9) while other variations occur among remaining links."

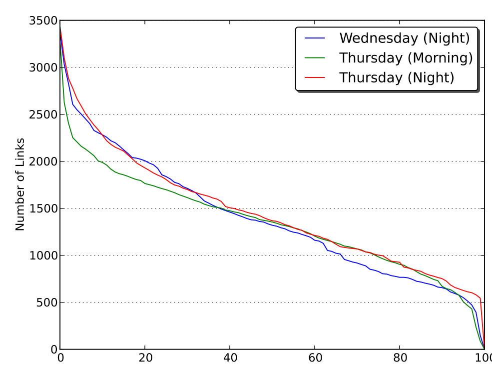
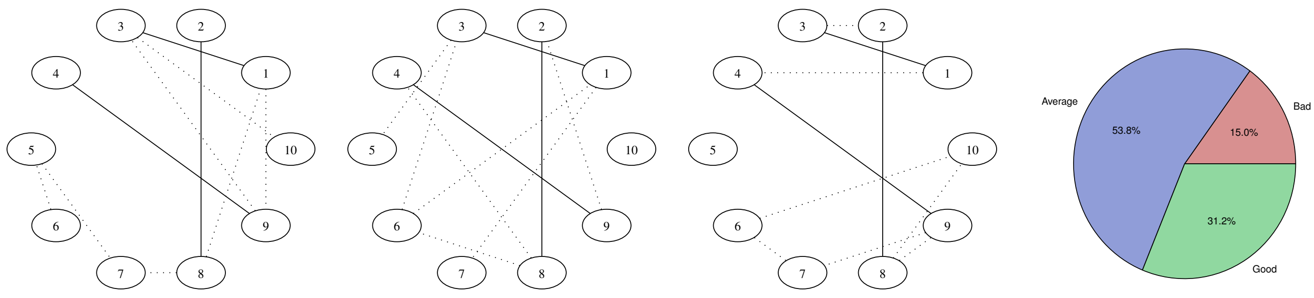


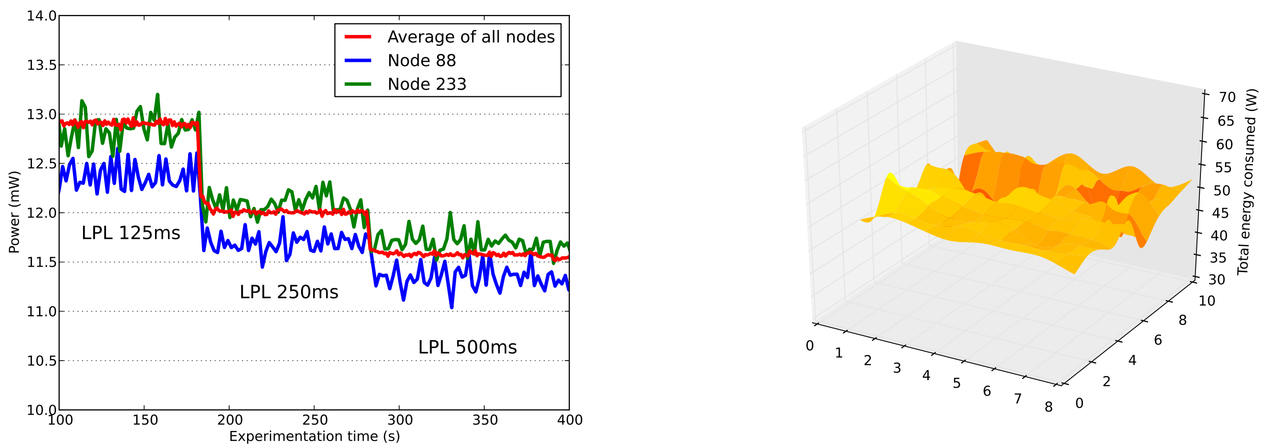
Fig. 5: Links quality over time.

As previously mentioned, in real deployments the link quality and symmetry may significantly vary over time. In Figure 5, a detailed representation of link quality throughout the whole experimental procedure is shown, in particular the number of bidirectional links per PRR is presented. As we can observe, both three experiments follow similar curves. In fact,



(a) Wednesday (Night) (b) Thursday (Morning) (c) Thursday (Night) (d) Links quality

Fig. 4: Links quality and stability over time.



(a) Average energy consumption of nodes. (b) Total energy consumption of nodes in 3D.

Fig. 6: The average and total energy consumption of nodes, in IoT-LAB testbed.

more than 500 bidirectional links have PRR above 90%. In order to have more general view of the links, we labeled the link qualities in three categories the *bad links* (where PRR is less than 10% in both sides), the *good links* (where PRR is over 90% in both sides) and finally the *average links*. Figure 4(d) presents the results of the link quality. Little more than 30% of the links are good links since they remain unchanged while more than 80% are average links. Yet, this behavior is very hard to reproduce in simulations. In order to get closer to real deployment, researchers often need to remove the assumption, while preserving identical density over time (in order to perform a meaningful evaluation). Thus, we evaluated the density of neighbors over time in order to verify if this property was fulfilled in IoT-LAB. The Table III presents the results. As it is shown, the IoT-LAB testbed presents stability since the density is about 14 neighbors where the links among these nodes have PRR above 90% both in three experiments. Then, in IoT-LAB, the density criterion remains stable while the links are changing. It thus allows researchers to remove assumptions one at a time, to best evaluate their protocols. It also shows that the testbed can be used either way. Indeed, it can be used as a scientific tool, producing scientific results that can be reproduced and slightly vary over time, even with a changing topology of identical density. When enlarging the set of considered links for the experimentations, users would introduce more and more randomness due to the lower quality of those links, thus reaching real life conditions and meeting some of the requirements for the validation of a proof-of-concept and iterative prototyping.

TABLE III: Average number of 1-hop neighbors over the time

Date	Nbr of Neighbors	Confidence Interval
Wednesday (Night)	13.95	2.16
Thursday (Morning)	13.38	1.9
Thursday (Night)	15.34	1.94

c) Energy performance: as mentioned in section II, simulators can only provide an estimation of the energy consumption of sensor nodes. Indeed, they often provide a linear model, where all nodes follow the same smooth and predictable energy consumption pattern, and only consider a subset of energy consumption causes only (e.g. nor sensors or memory). On the contrary, in reality two identical nodes may follow different consumption patterns, and some little changes in the protocol can have unexpected impact. In fact, two distinct nodes would have various energy consumption profiles, either they embed different component (e.g. memory, CPU). Or, even though they could come with two identical hardware configurations, it can also happen due to the electric components manufacturing, their fixation over the silicium board, and so on. In IoT-LAB, the gateway allows for precise monitoring of the energy consumption for each node. This information is directly provided to the user, as well as parser to compute the average and total energy consumption of the network (and in option of some specific nodes) as well as an energy map. In this experiment, we want to demonstrate how energy can be easily and efficiently retrieved in IoT-LAB. To do so, we considered the example of X-MAC duty-cycle parameter and we evaluated their energy impact. The results are depicted in Figure 6(a). Overall few differences among nodes, which is expected since they all embed the same hardware components. It is worth to point out that in the former case, energy profiles can be first established. Then, the end user would choose nodes having more or less the same profile in order to get significant results. In the latter, heterogeneous nodes can be picked in order to reflect what will be encountered when facing reality through real-life deployments.

We also performed a mapping of the energy depletion throughout the network, in order to identify its repartition and disparity among the nodes. To do so, we measured the total energy consumption of each node for the total duration of an experiment. The results of this evaluation are displayed in Figure 6(b). Looking more closely, the nodes are having heterogeneous behavior, meaning that there are differences in energy consumption (i.e. from 35 to $50W$) which can be seen as a drawback from testbeds. This difference should be due to the used hardware since, with the same setup in simulation the nodes would consume precisely the same. Hence, this anomaly brings us step towards real deployments where the researchers may have to deal with number of unexpected behaviors. Hence, this is the one of the main advantages of testbeds compare to simulators, as they allow us to get a flavor of real deployments without having to modify the code.

V. Conclusion and Future work

WSN simulation is an important step for WSN development. WSN simulators allow users to retain or simplify some assumptions to serve evaluation requirements (e.g. link stability, radio propagation) by tuning configurable parameters. On the other hand, a testbed is a platform for experimentation of development projects which allows for rigorous, transparent, replicable testing of new models, and reflects the reality that their solution would face during real deployments. Nowadays, both in simulations and testbeds, protocols, models, even new technologies can be evaluated at a very large scale. Testbeds and simulators are two complementary validation tools for deploying a successful real deployment experiment. Hence, once both of them provide successful results, engineers can start considering real deployments.

In this paper, we demonstrated how the IoT-LAB platform can be efficiently and successfully coupled with simulations. For instance, we exposed how both local and global energy consumption can be precisely monitored, and to what extent the link stability assumption can be removed or not, at the users choice. This way, the evaluation campaign of WSN protocols can go one step further towards real deployment by removing the above mentioned assumptions, at little time cost and with limited complexity.

As a future work, our vision is to further explore the role of testbeds in the protocol or in the model development procedure. We would like to study whether the testbeds are minding or bridging the gaps between simulations and real deployments, and even more if testbeds should stick to simulations or prepare to real deployments. Hence, first we will investigate if with monitoring tools and control channels of testbeds it is possible to identify these problems (i.e. minding). Furthermore, we plan to examine the possibility of reducing these gaps through testbeds (i.e. bridging).

References

[1] T. W. Hnat, V. Srinivasan, J. Lu, T. I. Sookoor, R. Dawson, J. Stankovic, and K. Whitehouse, “The hitchhiker’s guide to successful residential sensing deployments,” in *Proc. of ACM Sensys’11*, pp. 232–245.

[2] A. J. Jara, M. A. Zamora, and A. F. Skarmeta, “An internet of things-based personal device for diabetes therapy management in ambient assisted living (aal),” *Personal and Ubiquitous Computing*, vol. 15, no. 4, pp. 431–440, 2011.

[3] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, “Experimental evaluation of wireless simulation assumptions,” in *Proc. of ACM MSWIM’04*, pp. 78–82.

[4] S. Lohs, R. Karnapke, and J. Nolte, “Link stability in a wireless sensor network - an experimental study,” *Sensor Systems and Software*, vol. 102, no. 2, pp. 146–161, 2012.

[5] C. A. Boano, J. Brown, Z. He, U. Roedig, and T. Voigt, “Low-power radio communication in industrial outdoor deployments: The impact of weather conditions and atex-compliance,” *Sensor Applications, Experimentation, and Logistics*, vol. 29, pp. 159–176, 2010.

[6] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, “Sensorscope: Out-of-the-box environmental monitoring,” in *Proc. of ACM/IEEE IPSN’08*, pp. 332–343.

[7] A. Intana, M. R. Poppleton, and G. V. Merrett, “Adding value to wsn simulation through formal modelling and analysis,” in *Proc. of SESENA’13*, pp. 24–29.

[8] P. Levis, N. Lee, M. Welsh, and D. Culler, “Tossim: accurate and scalable simulation of entire tinyos applications,” in *Proc. of ACM SenSys’03*, pp. 126–137.

[9] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, “Cross-level sensor network simulation with cooja,” in *Proc. of IEEE LCN’06*, pp. 641–648.

[10] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, “The hitchhiker’s guide to successful wireless sensor network deployments,” in *Proc. of ACM SenSys’08*. pp. 43–56.

[11] K. Langendoen, A. Baggio, and O. Visser, “Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture,” in *Proc. of IEEE IPDPS’06*. , pp. 174–174.

[12] R. S. Dilmaghani, H. Bobarshad, M. Ghavami, S. Choobkar, and C. Wolfe, “Wireless sensor networks for monitoring physiological signals of multiple patients,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 5, no. 4, pp. 347–356, 2011.

[13] C. B. des Roziers, G. Chelius, T. Ducrocq, E. Fleury, A. Fraboulet, A. Gallais, N. Mitton, T. Noel, and J. Vandaele, “Using senslab as a first class scientific tool for large scale wireless sensor network experiments,” in *Proc. of IFIP Networking*, 2011, pp. 147–159.

[14] P. Levis, S. Madden, J. Polastre, R. Szewczyk, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, “Tinyos: An operating system for sensor networks,” in *in Ambient Intelligence*. Springer Verlag, 2004.

[15] A. Dunkels, B. Gronvall, and T. Voigt, “Contiki - a lightweight and flexible operating system for tiny networked sensors,” in *Proceedings of IEEE LCN’04*, pp. 455–462.

[16] H. Kdouh, H. Farhat, G. Zaharia, C. Brousseau, G. Grunfelder, and G. Zein, “Performance analysis of a hierarchical shipboard wireless sensor network,” in *Proc. of IEEE PIMRC’12*, pp. 765–770.

[17] M. Buettner, G. V. Yee, E. Anderson, and R. Han, “X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks,” in *Proc. of ACM SenSys’06*. pp. 307–320.