A Survey of WSN Testbeds Deployment

Pablo Godoy[1,4], Lucas Iacono[1,3,4], Ricardo Cayssials[2], Carlos García Garino[1,4,5]
[1]ITIC, Instituto universitario para las Tecnologías y las Comunicaciones, UNCuyo.
[2]Ingeniería Eléctrica y de Computadoras, Universidad Nacional del Sur.
[3]Facultad de Ingeniería, Universidad de Mendoza.
[4]LAPIC, Laboratorio de Investigación y Desarrollo para la Producción Integrada por Computadora, Carrera de Redes y Telecomunicaciones, ITU.
[5]Facultad de ingeniera, Universidad Nacional de Cuyo.
pgodoy@itu.uncu.edu.ar, lucas.iacono@um.edu.ar, ricardo.cayssials@uns.edu.ar, cgarcia@itu.uncu.edu.ar

***Abstract*—A WSN testbed is a WSN (wireless sensor network) deployed in a controlled environment, that users can use for testing new developments, such as communication protocols, applications, etc. Users can access remotely, configure or program nodes, retrieve experimental results, and in some cases, interact in real-time with the WSN nodes. This paper is a survey about different approaches used for building WSN testbeds. This information can be useful for building new WSN testbeds, or for developing improvements and solutions to current problems of WSN testbeds. The paper presents information about: architecture, hardware and software components, interaction with users, open issues and possible improvements.**

***Index Terms*—WSN testbeds, development of WSN, test of WSNs, WSN development tools**

I. INTRODUCTION

A. An Overview of WSNs

A wireless sensor network (WSN) is a system composed of devices known as nodes or motes with capacity of sensing, data processing, data storage, and wireless communication [1]. The WSNs have a large number of applications, such as: environmental monitoring, agriculture, health monitoring, factory and process automation, building automation, military applications, etc. It is expected that the number of applications, and the number of WSNs grow enormously [1][2][3][4].
WSNs have specific design requirements:

- Large number of nodes. This request imposes the need of low cost nodes, and therefore, reduced capabilities of processing and memory.
- Enough useful life, without or minimal maintenance. This request leads to two different requirements: (1) robustness and (2) low power consumption, due to nodes are powered by batteries. In adittion, nodes have to consume all the power from the batteries (see [5] about this requirement).
- Reliability (regarding the data integrity).
- Autonomy, a WSN must be able to adapt itself to topology changes due to relocation or damage of nodes, to solve failures, etc.

These specific design requirements and the growing number of applications have motivated researchers and companies to develop hardware and software platforms for WSN.

B. An Overview of Performance Evaluation Tools for WSN

The development of applications, hardware and software for WSN requires performance evaluation tools. Common performance evaluation tools are: computer simulation and emulation, testbeds and real deployments [6].

- Simulators and emulators: are software tools that can imitate the behavior of a real system or certain parts of a real system (see [6][7][8] for details).
- Real deployment: consists on a actual WSN that users deploy for performing a certain experiment.
- Testbed: A WSN testbed consists on an actual WSN deployed permanently, generally with public access, remotely accessible, that users can access for performing experiments, without having to deploy a WSN locally.

Section II-A presents differents papers that compare these performance evaluation tools.

C. An Overview of WSN Testbeds

WSN testbeds are deployed in a controlled environment, generally with public access and ongoing maintenance. It is a intermediate tool between a real deployment and a simulator or emulator. Unlike a simulator or an emulator, users can test their developments on actual hardware. Unlike a real deployment, users do not need buy sensor nodes, deploy them in the field, implement tools for monitoring and data analysis, etc., due to a WSN testbed is a real deployment that usually has these elements.
The WSN testbed architecture can be divided in two parts:

- The system under test: A WSN on that run the users experiments.
- The management system: enables to program and monitor the nodes, remote access, retrieve experiment data, users management, etc.

Existing WSN testbeds have different architectures. There are common components, but also there are components that are not common to all WSN testbeds. This paper analyzes different architectures for deploying WSN testbeds. For this purpose, we analyze different approaches about the following elements:

- Nodes (the system under test).
- Back-end.
- Operation in batch mode and in real-time mode

- Injection of data
- Data storage
- WSN testbeds without remote access
- In situ analysis tool for WSN testbeds
- Mobile nodes and localization system
- Virtual mobility for static WSN testbeds
- Miniaturized and full scale WSN testbeds
- Distributed testbed
- Web interface

D. Organization of This Paper

The rest of the paper is organised as follows: Section II presents related works. Different components of a WSN testbed and different alternatives for building these components are discussed in section III. Then open issues and possible improvements are presented in section IV and concluding remarks are provided in section V.

II. RELATED WORKS

A. Comparison of Performance Evaluation Tools for WSN

This section cites results obtained by several authors that comparing the different kinds of performance evaluation tools for WSN: simulation and emulation, testbeds, and real deployments.

Real deployments produce the more accurate and reliable results. However a real deployment requires considerable financial resources and considerable time, due to it requires programing and reprograming many nodes, deploy the nodes, to implement tools for extracting performance data, etc. [9]. Therefore, most researchers choose simulation tools. However, a simulator and a real deployment produce different results, due to different reasons, some of these are listed below:

- Simplifications and unrealistic assumptions in the models used for designing simulators [4][6][9][10].
- Behaviors very difficult to model, such as: cross-layer interactions, external interferences, environmental behavior, errors caused by hardware or drivers, etc. [9][10].
- Simulators do not enable to control parameters such as: precise packet timing, hardware interrupts, and real events in physical and MAC layers [11].

There are few experiment results that compare testbeds and simulators for WSN. Below some papers are cited. Elhadi M. Shakshuki et al. [12] conducted a experiment for highlighting the differences between the results produced by a simulator and a real deployment. The experiment consisted in to run a MAC protocol for WSNs on two simulators (NS-2 simulator and a customized simulator) and on a real deployment. Some results are:

- The simulations showed lower energy consumption than real deployment.
- The phenomena of network segmentation and communication holes (both phenomena caused when many nodes stop working) occurred in the real deployment, but were not reproduced in the simulations.
- The drift of the simulation results compared to real deployment results depend on the topology used.

Langendoen [13] conducted a similar experiment on two testbeds (MoteLab [14] and MistLab [15]), using TinyOS for programming the nodes of both testbeds, and a simulator, TOSSIM [16], the TinyOS simulator. The obtained results (which are not listed for brevity, see [13] for details) led to Langendoen to state: ”The results from the TOSSIM simulator differ dramatically from those observed on the physical testbeds”.

These studies show that a development or application can be proved in a simulator for obtaining approximate results. But for knowing the effect of the actual environmental conditions or the hardware behavior, real deployments are needed [17].

B. Others surveys about WSN testbeds

Other surveys about WSNs mention briefly existing testbed, but with few details. Information about WSN testbeds can be found in [1][4][6][18][19].

Steyn and Hancke [20] present the most complete list about the available WSN testbed at the date, highlighting the outstanding features of each testbed. Steyn and Hancke [20] also present a comparison table and a classification of WSN testbeds. This classification divides the WSN testbeds, according to the implementation of the management system, in the following types: testbeds of central server, testbeds of single PC and hybrid testbeds. According other criteria, Steyn and Hancke divide WSN testbeds in: testbeds of specialised hardware (testbeds built with specialised hardware that enables specific functions), multi-site testbeds, testbeds with in-band management (without back-channel), and testbeds for testing industrial applications.

Unlike these papers, our paper does not present a summary or list of existing WSN testbeds, but presents in depth how they are built.

III. CONSTRUCTION OF WSN TESTBEDS

This section presents the differents design variants used for building WSN testbeds.

A. Nodes

Nodes that comprise some WSN testbeds are not equipped with any additional hardware. In other WSN testbeds, these nodes have a interface with the testbed, for example Ethernet board (eg. [14]), serial to Ethernet converters (eg. [11]), etc. In other cases, the nodes have sensors (eg. [21]) and measuring instruments (eg. [14][21]). A special case is Kansei testbed [22], whose nodes are formed by different WSN nodes, a single-board computer and sensors.

B. Back-End

The interconnection system between nodes and the rest of the system, called back-end by several authors, enables communication between nodes and the testbed. The back-end has different functions, among these:

- Upload to nodes the programs that users need to test.
- Transmit the data produced by nodes during the experiment for being stored in databases.

- In testbeds that enable real-time interaction between users and nodes, the back-end enables this interaction.
- Perform monitoring and maintenance operations.

There are four approaches for implementing the back-end:

1) Ethernet: For example: w-iLab. T [10], WSNTB [11], MoteLab [14]. It has the disadvantage that usually typical sensor nodes do not have support for Ethernet, as a result, additional hardware is required, such as Ethernet boards attached to the nodes, serial to Ethernet or USB to Ethernet converters.

2) USB cables: For example: NetEye [18][23], Indriya [8]. Most of the commercial nodes have support for USB. However, the maximum cable length is limited to 5 meters. As a result, additional components are required for reaching larger lengths, such as USB hubs or computers. These WSN testbeds are formed by clusters of nodes. Inside each cluster, nodes communicate through USB cables, and the clusters communicate with the testbed by different means such as USB, Ethernet or IEEE802.11.

3) IEEE802.11: For example: Senseiuu [24]. Every node must support this type of wireless communication, in addition its normal communication channel. Rensfelt et al. [24] evaluated the interference between IEEE 802.11 and 802.15.4 signals. The authors conclude that is needed to take some precautions for avoiding interference (see [25] for details).

4) The 802.15.4 channel: For example: X-Sensor [26]. The WSN's normal communication channel is shared between normal operation and programing tasks. This method is the most economical and simple. However it makes impossible real time communication with the nodes, due to typical delays in a WSN.

C. Operation in Batch Mode and in Real-Time Mode

During the time interval assigned to an user, a testbed can work in batch mode or real-time mode, or both (this terminology varies according to the author).

In batch mode, after programming the nodes, the experiment starts running automatically, and the data generated during the experiment are stored in a database. In this mode, the user can not interact with her/his experiment until the same has ended. In general WSN testbeds use the batch mode.

In real time mode, the user can interact with nodes while the experiment is running. Some WSN testbeds that use real-time mode are MoteLab [14] (that gives it the name of "real-time access") and WSNTB [11] (that gives it the name "local mode"). In order to achieve this interaction, the chosen method by these WSN testbeds is a TCP connection. Through these connections, data is transmitted in both directions.

D. Injection of Data

Some testbeds enable users to inject data to nodes. This enables to test the nodes behavior when they receive messages from the outside, and on the other hand, this functionality can be a useful tool for debugging tasks. These data must be predefined and preloaded on a file. This file must be sent to the testbed when user creates the experiment. (eg Neteye [18][23]). It should be noted that this is not real-time interaction.

E. Data Storage

In order to store data, most testbeds cited in this paper use a MySQL database. In general, information stored in this database comprises: user information, node status, overall testbed status, information about pending experiments (executables to be loaded on each node, experiments configurations, etc.), results produced by experiments (except Kansei [22], that stores experiment results in the server filesystem), etc. Users can access these database, but with restricted access rights.

F. WSN Testbeds Without Remote Access

Not all WSN testbeds have remote access. Some WSN testbeds have been thought as replicable and standard platforms, not as remote laboratories. The objective of these replicable platforms is that users can perform experiments in different places of the world using a common platform. These platforms are designed with commercially available hardware and accessible software. Some examples are [21] and [24].

G. In Situ Analysis Tool for WSN Testbeds

There are tools for evaluating the performance of a WSN testbed or a WSN in general, through the evaluation of different parameters such as connectivity between nodes, packet delivery rate, received signal strength indicator (RSSI), link quality indicator (LQI), etc. This section presents some of the existing tools.

1) MoteLab Connectivity Daemon: MoteLab Connectivity Daemon is an application used for evaluating connectivity among nodes in MoteLab testbed [14][27]. It runs periodically when the testbed is free of tasks, and works like any other experiment. It operates as follows: divides the time into cycles, and in each cycle, only one node transmits data packets, while the others nodes listen for receiving these data packets. When a node receives one of these packages, it records this event and the value of RSSI. When the work ends, a Perl script uses the data obtained for calculating packet loss rates between each pair of nodes. This information is used to graphically illustrate the connectivity among nodes through the web interface. Users can retrieve these data.

2) TestbedProfiler: TestbedProfiler is an analysis tool for WSN testbeds and WSN in general. It can be used in WSN whose nodes have the Texas Instruments CC2420 radio (eg MicaZ and TelosB) [28]. The WSN has to have a back-channel for sending data to the central server. TestbedProfiler analyzes the following parameters: connectivity between nodes, packet delivery rates, average neighbor counts, typical received signal strength indicator (RSSI) and link quality indicator (LQI).

TestbedProfiler consists of three software components:

- a TinyOS application running on the nodes.
- a central server.
- post-experiment analysis scripts.

The TinyOS application that runs on the nodes performs the following tasks: sends broadcast packets, of different sizes and power levels, and listens for receiving these packets. At one point, only one node can send these broadcast packets, while the others listen. The central server coordinates the nodes. When a node receives a data packet, sends to the central server a message reporting the following information: its ID, the node that sent the packet, the RSSI, the LQI, the sequence number, the payload size, and the transmitted power level. The central server stores this information.

The post-experiment analysis scripts (written in Python) analyze the information produced, and generate statistics and graphical representations of this information to the user.

H. Mobile Nodes and Localization System

Very few WSN testbeds have mobile nodes. Some of these are: [29][24] and [25], however, these do not have remote access. Mobile Emulab [29] was a remotely accessible WSN testbed with mobile nodes, but its website states that it is not longer supported. In all cases the testbeds use robots for transporting nodes. Robots can follow entirely free paths such as in [29] and [25], or predefined paths such as [24].

Users need to know the localization of the mobile nodes, so the testbeds must have a localization system. There are two approaches for localization system:

- use cameras (eg [29] and [25]).
- through marks on the floor along with the counting of the wheels revolutions (eg [24]).

The use of cameras is more accurate and does not limit the nodes mobility (except to the coverage area of the cameras), but require the use of image recognition algorithms. The second approach is subject to the error of the revolution counting systems and nodes have to pass through the marks, which may limit the nodes mobility.

I. Virtual Mobility for Static WSN Testbeds

A major shortcoming of the current WSN testbeds, mentioned by several authors, is the lack of mobility [24][25]. Puccinelli and Giordano [30] propose a solution based in virtual mobility called ViMobiO. The authors propose the use of logical nodes, that can move through the network, being the actual nodes (ie physical nodes) possible locations for logical nodes. The authors use the wireless channel for performing the opetarions that enable the movement of logical nodes.

In summary, it works by dividing the time in periods, and every period is divided into two. In a part of a period, the nodes work according to their normal operation, and in the another part of the period, the nodes perform the tasks related to virtual mobility.

ViMobiO was implemented in TinyOS and was tested on MoteLab testbed.

J. Miniaturized and Full Scale WSN Testbeds

According to the classification mentioned in [31], WSN testbeds may be full scale or miniaturized. Full scale testbeds (or real size testbeds) are closer to the actual environmental conditions, but require large size facilities for deploying the nodes. In a miniaturized testbed, nodes are placed to smaller distance than a real size WSN, even just meters or centimeters. For simulating nodes placed to actual distance, miniaturized testbeds use different approach:

- Attenuators (eg [17][22][31]).
- Discard data packets coming from nodes placed to a larger distance than a specific threshold (eg [25]).

K. Distributed Testbed

There are some WSN testbeds that have distributed architecture, with facilities deployed in locations geographically separated. Some examples are: X-Sensor [26], WISEBED [4], SensLAB [32]. WISEBED uses the concept of ”virtual links”. These virtual links enable to create network topologies using nodes that are on different testbeds. This is accomplished using a software component, housed in the gateways, that controls the flow of data between physically separated testbeds.

L. Web Interface

Most of WSN testbeds cited in this paper use a web interface for communicating with the users. Through this web interface, users can program nodes, schedule experiments, retrieve experiment results, interact in real time with nodes, see pictures and schemes, access to manuals, etc. Most of WSN testbeds use PHP and Javascript programming languages. An exception is TWIST [17], that does not have web interface. In TWIST, the nodes must be programmed through Python scripts, that are invoked remotely via ssh.

M. Central Server and Management Software

A computer system play the role of the central server that enables the remote access. The database is hosted at central server, where management software is partially installed. The management software controls the experiments execution and user activity, programs the nodes and controls others components. This management software has a distributed implementation, with component on the central servers, on gateways or intermediate computers, and on the nodes.

N. Summary

Table I summarizes the characteristics of the WSN testbeds mentioned in this section.

IV. OPEN ISSUES AND POSSIBLE IMPROVEMENTS

Several authors cite problems and possible improvements for current WSN testbeds, and we have detected others, which are presented below.

A. Different Results for a Same Experiment

Koen Langendoen [13] shows that different testbeds produce different results for a same experiment. Langendoen states that every testbed is unique, due to they use different type of nodes, different number of nodes, different network topologies, etc. As a result, to reproduce results obtained from a WSN testbed in other WSN testbed is very difficult, maybe impossible. In addition, the results produced by a WSN testbed can

TABLE I
SUMMARY OF FEATURES

Nodes	Constitute the system under test
Back-end	Ethernet, USB cables, IEEE802.11 or the 802.15.4 channel
Interaction with users	Batch mode, real-time mode or injection of data
Data storage	MySQL or server filesystem
In situ analysis tool	MoteLab connectivity daemon or Testbed-Profiler [28]
Mobility	Few WSN testbeds have mobile nodes. Mobility can be real or virtual
Deployment	Distributed or centralized
Web interface	PHP and Javascript
Management software	Distributed implementation

be difficult to reproduce on the same WSN testbed, due to changes in topology of the WSN produced by death and resurrection of nodes, and the existence of the gray area (part of the transmission range with a irregular reception pattern). The paper of Langendoen concludes with the phrase ”comparing results from different testbeds is like comparing apples and oranges” [13].

B. Single Hardware Platform

Another unresolved problem cited by Ze Zhao et al. [19] is that current WSN testbeds always use a single hardware platform, and this hardware platform is not flexible. This limits the results and conclusions of a experiment to the features of the hardware used.

C. Need for More WSN Testbeds

According with [4],more publicly available heterogeneous WSN testbeds are necessary. As a result, several authors encourage the deployment of new WSN testbeds, and provide to users the software components of their own WSN testbeds (eg MoteLab [14] and WISEBED [4]).

Some WSN testbeds enable that users associate other WSN testbeds to their infrastructure (eg WISEBED [4]). For this purpose, the child testbed must meet certain requirements imposed by the father WSN testbed (see [4] for details).

D. Lack of Outdoor or Industrial WSN Testbeds

Many WSN applications require outdoor deployments, so, new developments should be tested outdoor for a more accurate evaluation. But none of the WSN testbeds cited in this paper have outdoor facilities.

There is only one WSN testbed designed for industrial environments [31].

E. Need of Advanced Programming Skills

Programming TelosB family nodes requires advanced programming skills (NesC language), as a result, users without this knowledge can not be able to use these testbeds. Most of testbeds cited in this paper use nodes that must be programmed with C or nesC language [33]. An interesant option is to deploy WSN testbeds with nodes easy to program, like the XBee nodes [34].

F. Nodes that Are not Powered by Batteries

Nodes of almost all current testbeds are powered by wall sockets, USB or other non-finite source (not batteries), while in normal field conditions, nodes are powered by batteries. The use of non-finite source power is a deviation from the actual field conditions. The lifetime of a node in a WSN is determined largely by the life of their batteries, so that the batteries discharge is a very important parameter.

Only one of the cited WSN testbeds, SensLAB [28], uses batteries (enables users to select between batteries or DC power). These batteries are recharged when the nodes are powered from DC.

Other solution might be to use a mathematical model for simulating the behavior of batteries during the experiment.

G. Can not Make Long-term Tests

Shor term experiments, a couple of hours for instance, have been reported in the literature [21]. However in practice researchers can require long term experiments, ie. weeks or months.

Günes, M. et al. [21] present information on several interesting WSN deployments and WSN testbeds built in the past. An important conclusion cited in the paper is the lack of information about long term experiments. Among other missing information neither results nor problems are reported for several months experiments. For example data are listed only for short intervals of time or the frequency of problems reported is not enough. Then the authors emphasize the need for long-term experiments.

V. CONCLUSION

In general, WSN testbeds have similar components: WSN nodes, back-end system, a central server, database, and web interface. There are similarities and differences on the design of these components. Every component may be designed in different ways, and every solution has advantages and disadvantages. For example, adding hardware to nodes enables additional functionalities, such as measuring current, but this leads to a higher cost. It is not possible to build a WSN testbed that simultaneously optimizes cost and deployment size, cost and mobility, low maintenance and environment realism (an outdoor deployment is a more realistic environment than a laboratory deployment, but require more maintenance), ease of use and flexibility, etc. It is hard to build a WSN testbed that overcomes all problems mentioned in this paper without incurring a high cost. Each designer must decide which objectives to prioritize.

Different testbeds produce different results for a same experiment, and even one testbed produces different results for a same experiment. Therefore, if a user need to test a new application, for example, a new communication protocol, must use several and different WSN testbeds. This will ensure that the new application can run properly under different network configurations, different environments, different hardware, etc. As a result, more publicly available heterogeneous WSN testbeds are necessary, with different configurations, especially for outdoor and industrial environments, and for long-term experiments.

ACKNOWLEDGMENT

The authors acknowledge the financial support received from the Instituto de Desarrollo Industrial Tecnolgico y de Servicios (IDITS) and the Agencia Nacional de Promocin Cientfica y Tecnolgica (ANPCyT) through the project PID-PAE 146 Grid Embebida y de Sensores. Pablo Godoy and Lucas Iacono thank doctoral fellowships awarded by the ANPCyT through the program PRH funded by the ANPCyT and UNCuyo, and project PID PAE-146 respectively.

REFERENCES

[1] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer Networks*, vol. 52, no. 12, pp. 2292 – 2330, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128608001254

[2] T. Arampatzis, J. Lygeros, and S. Manesis, “A survey of applications of wireless sensors and wireless sensor networks,” in *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, june 2005, pp. 719 –724.

[3] Q.-D. Ho and T. Le-Ngoc, “A wireless sensor network testbed,” in *Communication Networks and Services Research Conference (CNSR), 2010 Eighth Annual*, may 2010, pp. 304 –309.

[4] H. Hellbruck, M. Pagel, A. Kroller, D. Bimschas, D. Pfisterer, and S. Fischer, “Using and operating wireless sensor network testbeds with wisebed,” in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean*, june 2011, pp. 171 –178.

[5] J. Polastre, R. Szewczyk, and D. Culler, “Telos: enabling ultra-low power wireless research,” in *Proceedings of the 4th international symposium on Information processing in sensor networks*. IEEE Press, 2005, pp. 48–es.

[6] M. Imran, A. Said, and H. Hasbullah, “A survey of simulators, emulators and testbeds for wireless sensor networks,” in *Information Technology (ITSim), 2010 International Symposium in*, vol. 2, 2010, pp. 897 –902.

[7] C. Singh, O. Vyas, and M. Tiwari, “A survey of simulation in sensor networks,” in *Computational Intelligence for Modelling Control Automation, 2008 International Conference on*, dec. 2008, pp. 867 –872.

[8] M. C. C. M. Doddavenkatappa and A. L. Ananda, “Indriya: A low cost, 3dwireless sensor network testbed,” in *TRIDENTCOM 2011*, 2011.

[9] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, “Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols,” in *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 3. IEEE, 2005, pp. 1664–1669.

[10] S. Bouckaert, W. Vandenberghe, B. Jooris, I. Moerman, and P. Demeester, “The w-ilab. t testbed,” in *TridentCom 2010: proceedings CD of the 6th International conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, 2010.

[11] J.-P. Sheu, C.-J. Chang, C.-Y. Sun, and W.-K. Hu, “Wsntb: A testbed for heterogeneous wireless sensor networks,” in *Ubi-Media Computing, 2008 First IEEE International Conference on*, 31 2008-aug. 1 2008, pp. 338 –343.

[12] E. M. Shakshuki, H. Malik, and T. R. Sheltami, “A comparative study on simulation vs. real time deployment in wireless sensor networks,” *Journal of Systems and Software*, vol. 84, no. 1, pp. 45 – 54, 2011, information Networking and Software Services. [Online]. Available: http://www.sciencedirect.com/science/article/B6V0N-50X2N9M-3/2/0add55e8a589d148a17988f5821cf751

[13] K. Langendoen, “Apples, oranges, and testbeds,” in *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, oct. 2006, pp. 387 –396.

[14] G. Werner-Allen, P. Swieskowski, and M. Welsh, “Motelab: a wireless sensor network testbed,” in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, april 2005, pp. 483 – 488.

[15] Mistlab website. [Online]. Available: http://mistlab.csail.mit.edu/

[16] P. Levis, N. Lee, M. Welsh, and D. Culler, “Tossim: accurate and scalable simulation of entire tinyos applications,” in *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM, 2003, pp. 126–137.

[17] V. Handziski, A. Köpke, A. Willig, and A. Wolisz, “Twist: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks,” in *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, ser. REALMAN ’06. New York, NY, USA: ACM, 2006, pp. 63–70. [Online]. Available: http://doi.acm.org/10.1145/1132983.1132995

[18] D. Sakamuri. (2008) Neteye: A wireless sensor network testbed. (master’s thesis). Wayne State University. Detroit, Michigan. [Online]. Available: http://www.cs.wayne.edu/ hzhang/group/publications/NetEye-thesis.pdf

[19] Z. Zhao, G.-H. Yang, Q. Liu, V. O. Li, and L. Cui, “Easitest: A multi-radio testbed for heterogeneous wireless sensor networks,” in *Wireless Sensor Network, 2010. IET-WSN. IET International Conference on*, nov. 2010, pp. 104 –108.

[20] L. Steyn and G. Hancke, “A survey of wireless sensor network testbeds,” in *AFRICON, 2011*, sept. 2011, pp. 1 –6.

[21] M. Gnes, B. Blywis, and J. Schiller, “A hybrid testbed for long-term wireless sensor network studies,” in *Int. Workshop on Sensor Network Engineering (IWSNE)*, 2008.

[22] E. Ertin, A. Arora, R. Ramnath, M. Nesterenko, V. Naik, S. Bapat, V. Kulathumani, M. Sridharan, H. Zhang, and H. Cao, “Kansei: a testbed for sensing at scale,” in *Information Processing in Sensor Networks, 2006. IPSN 2006. The Fifth International Conference on*, 0-0 2006, pp. 399 –406.

[23] W. state University. Neteye, wireless sensor network experimental facility. [Online]. Available: http://neteye.cs.wayne.edu/neteye/home.php

[24] O. Rensfelt, F. Hermans, P. Gunningberg, and L. Larzon, “Repeatable experiments with mobile nodes in a relocatable wsn testbed,” in *Distributed Computing in Sensor Systems Workshops (DCOSSW), 2010 6th IEEE International Conference on*, june 2010, pp. 1 –6.

[25] T. P. Lambrou and C. G. Panayiotou, “A testbed for coverage control using mixed wireless sensor networks,” *Journal of Network and Computer Applications*, no. 0, pp. –, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804511001123

[26] A. Kanzaki, T. Hara, Y. Ishi, N. Wakamiya, and S. Shimojo, “X-sensor: A sensor network testbed integrating multiple networks,” in *Complex, Intelligent and Software Intensive Systems, 2009. CISIS ’09. International Conference on*, march 2009, pp. 1082 –1087.

[27] H. University. Motelab. harvard sensor network testbed. [Online]. Available: http://motelab.eecs.harvard.edu/index.php

[28] C. Metcalf, T. Camp, M. Colagrosso, and O. Chase, “Testbedprofiler: A validation tool for wireless sensor network testbed deployment,” in *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, oct. 2010, pp. 188 –191.

[29] U. of Utah. Mobile emulab. [Online]. Available: http://www.emulab.net/tutorial/mobilewireless.php3

[30] T. Baumgartner, I. Chatzigiannakis, S. P. Fekete, S. Fischer, C. Koninis, A. Krller, D. Krger, G. Mylonas, and D. Pfisterer, “Distributed algorithm engineering for networks of tiny artifacts,” *Computer Science Review*, vol. 5, no. 1, pp. 85 – 102, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574013710000535

[31] J. Slipp, C. Ma, N. Polu, J. Nicholson, M. Murillo, and S. Hussain, “Winter: Architecture and applications of a wireless industrial sensor network testbed for radio-harsh environments,” in *Communication Networks and Services Research Conference, 2008. CNSR 2008. 6th Annual*, may 2008, pp. 422 –431.

[32] S. Consortium. Senslab: Very large scale open wsn testbed. [Online]. Available: http://www.senslab.info/

[33] D. Puccinelli and S. Giordano, “Vimobio: Virtual mobility overlay for static sensor network testbeds,” in *World of Wireless, Mobile and Multimedia Networks Workshops, 2009. WoWMoM 2009. IEEE International Symposium on a*, june 2009, pp. 1 –6.

[34] Z. Alliance. Zigbee specification. [Online]. Available: http://www.zigbee.org