View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by El Servicio de Difusión de la Creación Intelectual

A Cloud Based WSN Remote Laboratory for User Training

Pablo Godoy[1], Ricardo Cayssials[2], Carlos García Garino[3]
[1]ITIC, FCEN and Facultad de Ingeniería, Universidad Nacional de Cuyo, Argentina
[2]Universidad Tecnológica Nacional - FRBB, Argentina
[3]ITIC and Facultad de Ingeniería, Universidad Nacional de Cuyo, Argentina
pgodoy@itu.uncu.edu.ar,rcayssials@frbb.utn.edu.ar,cgarcia@itu.uncu.edu.ar

Abstract
This paper presents a WSN remote laboratory prototype aimed at user training. The prototype has been deployed on a Cloud Computing infrastructure, in order to take advantage of the benefits that this technology can provide to remote laboratories. An analysis of the state of the art allows to specified the features required for a remote laboratory intended to be applyed in user training. In this paper, the main results of this analysis are presented. Then, the implementation of the main modules that compose the remote laboratory, and the analysis that led to chose a deployment based on Cloud Computing are presented.
Some proof of concept experiments to test the applicability to user training were performed and the results are presented in this paper.
Keywords: Remote laboratory, user training, WSN, Cloud Computing.

1 Introduction
Remote laboratories are platforms that allow remote access to different types of equipments, devices, laboratories, etc. through Internet. They are complex systems that include a large number of components. They may be aimed at scientific research, application development, training, teaching, etc.
They are mainly composed of two systems:

- The system under test (equipments, devices, physical laboratory, etc.).
- The remote access system, that allows users to access the system under test to remotely perform experiments.

Remote laboratories have the potential of becoming an important tool for teaching through MOOCs (massive open online courses) [1, 2].
WSNs consist of devices, named nodes, having capability to sense variables, process and store data, and wireless communications. A WSN may include tens or hundreds of nodes [3]. There exist a great number of applications for WSNs, including: environmental monitoring, health monitoring, building automation, military, agriculture, etc [4, 5].
Cloud Computing is a model for allowing on demand and ubiquitous access to a configurable, virtualized and shared computational resources set (for example networks, servers, storage, applications and services), that can be quickly provided and released with minimal management effort and interaction with the Cloud services provider. These resources are offered by Cloud Computing services providers to external customers through Internet as web services, based on negotiated agreements between the service provider and customers [6, 7]. There are three models of Cloud Computing services:

- Software as a service (SaaS): The provider offers complete applications as a service through the Internet.
- Platform as a service (PaaS): The provider offers capacity for deploying and running applications created or acquired by users on the Cloud infrastructure.
- Infrastructure as a service (IaaS): The provider provides virtualized storage resources, servers, network

equipment, etc.

An open issue is to provide features like quality of service, security and reliable data storage to remote laboratories [8]. These requirements become more important in remote laboratories where student needs to be monitored and evaluated by teachers. Currently, providers of Cloud Computing services offer features that may solve the open issue above mentioned about remote laboratories, that is: high levels of service quality, security and reliable data storage (backups, intrusion protections, etc). Therefore, implementing some components of remote laboratories through Cloud Computing services can provide these features to remote laboratories.

This paper describe the construction of a Cloud based WSN remote laboratory prototype aimed at training activities and academic experimentation. The rest of the paper is organized as follow. Section 2 summarizes and analysed the state of the art. Section 3 describes special features that a remote laboratory requieres to be applyed in user training. Section 4 presents the implementation of the remote laboratory components. Section 5 describes proof of concept experiments. Final conclusions are drawn in section 6.

2 Related Works

In recent years, a large number of WSN remote laboratories have been developed. The papers of Steyn and Hancke [9] and El-Darymli et al. [10] are the most complete state of the art studies about WSN remote laboratories found in the literature. Others papers that include state of the art studies on WSN are: [11][12].

There are few and very recent papers about the integration of remote laboratories to Cloud Computing environments [13][14].

There are some papers about the integration of WSN to Cloud Computing environments [15][16][17]. Usually these papers present mechanisms to share data sensed by WSNs through Cloud Computing technologies.

Unlike papers mentioned above, this paper proposes a mechanism for the implementation of a WSN remote laboratory using Cloud Computing services.

3 WSN Remote Laboratories for Training Activities

3.1 Remote Laboratories Aimed to Training and Aimed to Research

A state of the art study allows to difference two types of remote laboratories:

- Remote laboratories aimed at development and research
- Remote laboratories aimed at education and users training

The most important difference between these types of remote laboratories is the knowledge level about the under test technology or equipment requiered by the user. Users of remote laboratories aimed at research requiere a very high level of knowledge. These users can perform experiments without help of anyone else. Remote laboratories oriented to research allow users to access to laboratory resources without any restrictions. For example, a WSN remote laboratory oriented to research, as MoteLab [18], allows users to load user programs or operating systems into nodes without any restrictions.

Users of remote laboratories for education and training purposes require a varied level of knowledge. For instance, in Marianetti [19] an academic remote laboratory is propose for teaching programmable logic devices configurations. In this remote laboratory, users with very low knowledge are guided through predefined and restricted experimients. Usually students can not load any program into the remote laboratory, and they only are allow to add blocks of code or modify variables. Because of the low level of knowledge of students, academic remote laboratories requiere more robust protection systems to avoid the risks produced by misconfigurations. In addition, in remote laboratories aimed at teaching tasks may be needed to supervise student activities, to verify that they perform practical work and

pass exams. As a result, additional components for

Table 1: Differences between remote laboratories oriented to training and oriented to research

	Oriented to research	Oriented to training
Users	Researchers and developers	Users who need to be trained in the use of some technology
Level of knowledge of under test equipments	High	Intermediate or low
Access restrictions	Few or none, the user can perform any experiment	Important restrictions, predefined experiments
Need to define part of the experiments	Users need to load their experiments and configurations, usually from scratch	Users interact with predefined experiments.
Robustness of protection systems	Low, users have high knowledge	High, users with little knowledge can make mistakes
Required accuracy of results	Very high, scientific purposes	Not high, training purposes
Web interfaces	Design few elaborate or it does not have web interface	The web interface design should enable a simple, intuitive and didactic use
Data stored in databases	Experiment results	Experiment results, and in some cases, the activity performed by users to allow tracking of their activities

monitoring the user activities are requiered [20]. Another difference is the accuracy on the results. In the case of a remote laboratory oriented to training, high accuracy may not be necessary. Instead, a remote laboratory oriented to scientific research requires the greatest possible accuracy. Table 1 summarizes the most important differences.

3.2 Applicability of Current WSN Remote Laboratories to Training Activities

State of the art studies about WSN remote laboratories enable to know components, modules and features common to remote laboratories [9][10][11][12]. In addition, these state of the art studies enable to know what components, modules or features allow or hinder to use of WSN remote laboratories forteaching or training task. A list of modules that need an adequate design in order to use aremote laboratory for user training is presented below:

- Operation in batch mode or real time mode: It is preferred real-time interaction to allow greater interaction with the equipment under test.
- Data storage: It must store information on the activity performed by users, in order to follow them, especially if the remote laboratory is used in teaching tasks.
- Web interface: It should allow a simple, intuitive and didactic use of the remote laboratory.
- Analysis tools: They can be useful for users to understand the operation of the WSN.

4 Implementation of a Cloud Based WSN Remote Laboratory Aimed to Training Activities

Based on the analysis presented in the section 3, it is possible to know the characteristics that must fulfill a remote laboratory aimed at user training tasks. From this information, a WSN remote laboratory prototype was proposed aimed to user training activities. This section describes the components of the prototype proposed.

4.1 Cloud Based Implementation and Traditional Implementation

The first versión of the remote laboratory was not implemented in the Cloud, but using a traditional scheme for remote laboratories: a PC that controls the equipment under test and allows remote access via LAN. Subsequently, the reading of most recent papers [8, 13, 14]

motivates the analysis of the advantages and disadvantages of integrating the remote laboratory with Cloud Computing technologies. Some advantages of Cloud based remote laboratories are:

- High availability and robustness of modules implemented on the Cloud
- High Scalability
- High processing power and large data storage capacity

For this reasons it was decided to deploy the remote laboratory in the Cloud.

The two implemented prototypes, traditional implementation (Figure 1) and Cloud Computing implementation (Figure 2), allowed to compare the differences between them. It could be noted that the interconnection system is the only component that requires major modifications. System hardware components of the interconnection system for the traditional deployment consist of USB cables and development boards. Software components consist on the operating system drivers and a program written in C ++ language, which receives and sends data frames to and from the WSN via the USB port. The interconnection system architecture for the traditional remote laboratory is shown in figure 1.

In order to implement the remote laboratory management modules in the Cloud, the

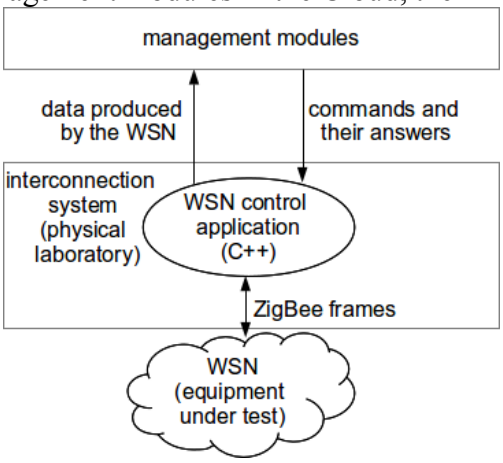

Figure 1: Interconnection system of the remote laboratory prototype based in a traditional implementation

interconnection system must interconnect the system under test with the management modules implemented in the Cloud. To achieve this objective, a client server application that allows to connect the equipment under test with the management modules deployed in the Cloud was implemented. Consequently, the interconnection system was divided in: (1) the Cloud, and (2) the physical laboratory, as shown in figure 2.

4.2 Cloud Computing Services Provider

The implemented system uses Cloud Computing services. There are several Cloud Computing service providers, and it is necessaryto choose one of them. The Cloud Computing service provider has to provide at least the following features:

- Data storage in the Cloud
- Support for communication with a remote application. This remote application is in charge of controlling the equipment under test. This application must be able to access the USB ports and the operating system of the remote computer
- Date and time services

In addition, the following requirement is not necessary, but it is desirable:

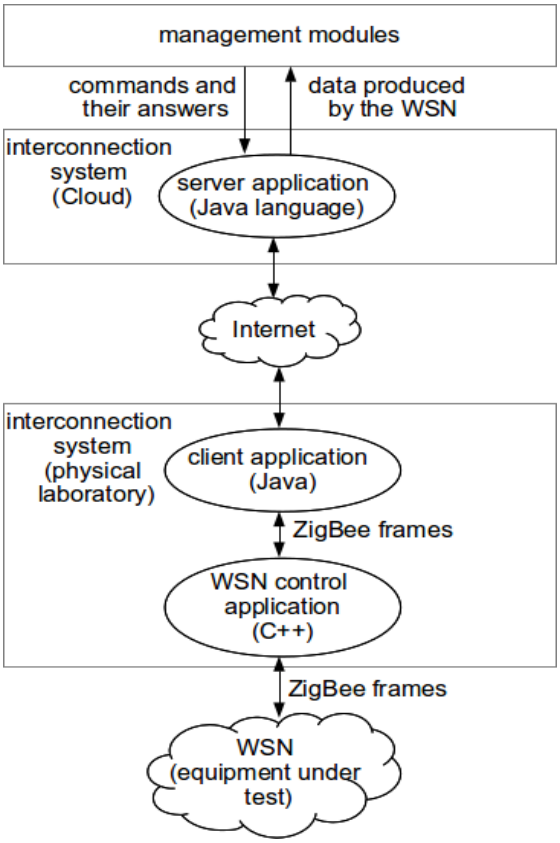

Figure 2: Interconnection system of the Cloud based remote laboratory prototype

- A mechanism for user authentication provided by the Cloud Computing service provider.

The first three above requirements are necessary to implement the remote laboratory. The service provider must enable to store data and should allow to consult the date and time to implement the user tracking system in the Cloud. In this way, data storage is secure and independent of the physical laboratory, so that it can work even if the physical laboratory is out of service. Communication with a remote application is necessary to implement the interconnection system with the equipment under test.

Regarding the user authentication service, it is preferable that the Cloud provides an user authentication service, which should be updated permanently to new types of attacks and vulnerabilities.

The required features from the Cloud Computing service provider are basic, since no high processing power or large space for data storage is needed. For this reason, it is expected that most Platform as a Service (PaaS) service providers can meet these requirements. For this first implementation, the Cloud Computing service provider chosen was Google App Engine (GAE) [21], that provides Platform as a Service (PaaS) services. These services allow users to create applications to run on Google infrastructure.

4.3 Interface Web

The web interface allows users to interact with the remote laboratory performing the following tasks:

- Reserve a time interval to access the remote laboratory
- Change and monitor configuration parameter values of the nodes
- Allow supervisor users to monitor the activity perfomed by other users

The web interface should be friendly, easy to use and intuitive. To build the web interface, a set of configuration parameters of the nodes under test have been selected. For each of these parameters, a set of possible values have been selected. These parameters are the same ones that users would program if they had the nodes connected to their computer. For example, for XBee nodes, if the user wants to program the transmit power, the PM (power mode) and PL (power level) parameters mustbe configured, and if user wants to set the reporting frequency data, the parameters ST (time before sleep), SP (cyclic sleep period) and SN (number of cycles to power down IO) should be programmed [22].

Nodo 0		Nodo 1		Nodo 2	
panID:	AAAA	panID:	AAAA	panID:	AAAA
ST:	nodata	ST:	0003E8	ST:	0003E8
SP:	nodata	SP:	000064	SP:	000064
SN:	nodata	SN:	000003	SN:	000003
SO:	nodata	SO:	0004	SO:	nodata
IR:	nodata	IR:	nodata	IR:	nodata
PM:	nodata	PM:	0001	PM:	0001
PL:	nodata	PL:	0004	PL:	0004
DB:	nodata	DB:	nodata	DB:	nodata
Pin valor configuracion		Pin valor configuracion		Pin valor configuracion	
DIO0:	No data x	DIO0:	No data E F	DIO0:	0204 ADC
DIO1:	No data x	DIO1:	No data x	DIO1:	1 Input
DIO2:	No data x	DIO2:	1 x	DIO2:	0 OutputL
DIO3:	No data x	DIO3:	1 OutputH	DIO3:	01EC ADC
DIO12:	No data x	DIO12:	1 OutputH	DIO12:	No data Disabled

Figure 3: Parameters configuration through the web interface

Figure 3 shows a screenshot of part of the web interface. For each node there is a list of settings parameters. In the case of the input/output pins, the user can configure their function (disabled, input, output, digital-analog converter) and visualize the value read for each pin. The web interface has been developed using HTML and JavaScript programming languages. These languages are accepted by most popular Internet browsers for both computers and mobile phones.
The web interface was implemented using Google Web Toolkit (GWT), a framework for creating web services running on a server and client applications that invoke these services [21]. These services are used to implement communication between users and the remote laboratory.

4.4 Other Modules that Composed the WSN Remote Laboratory

The main contributions of this work are in the above mentioned modules. However, a remote laboratory requires other modules. These modules are been built similar as the remotes laboratories mentioned in section 2, and are briefly mentioned below:

- Access control module: the access control is performed using the user authentication service provided by GAE. This service employs the Gmail user account and password in order to authenticate users.
- Reservation system: it is an application implemented with Java language that runs on the GAE infrastructure. All users that have a Gmail account can access to the remote laboratory and reserve a turn.
- User activity log module: Every action performed by users is stored in a log. Each log stores the date and time, the user name and the performed action. The logs are stored in the GAE database.
- Database: In order to access the database, the implementation of Java Data Objects (JDO) provided by GAE was employed. The database can store experimental results and users activities.

5 Proof of Concept Experiments

This section is intended to show some of the possible experiments that users can perform on the remote laboratory presented in this paper.

5.1 Response Time WSN

With this experiment the user can see how the configuration of a node affect its response time. The user can program the response time of the nodes by configuring the ST, SP and SN parameters, which have the following functions:

- ST: Inactivity time interval that must elapse before the node goes to low power mode.
- SP: Time interval that the node remains into low power mode.
- SN: Number of SP cycles during which the node remains in low power mode.

Figure 4 shows how the duty cycle of a XBee sensor node is configured through the ST, SP and SN parameters. The node can stay in two states: (1) active or (2) low consumption or sleep. The figure shows three different time intervals, which are:

- Active node performing tasks: When the node wakes up from sleep state and switches to the active state, tries to

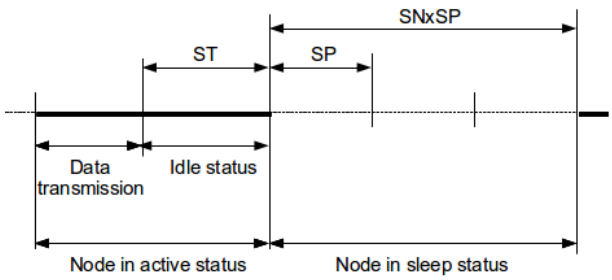

Figure 4: Duty cycle configuration of a XBee node

communicate with the coordinator node to receive queued messages, senses its input/output pins according its configuration, processes messages that the coordinator has sent, and sends to the coordinator the information required (including the readings of its pins).

- Active node not performing any task: Once the node has finished of performing all its tasks, waits a ST time interval for new communications with the coordinator node. If there are no messages from the coordinator node during the ST time interval, the node switches to low power consumption mode.
- Node in low power consumption state: the node stays in low power consumptiom mode for a SNxSP time interval before returning to the active state.

When the user sends a command to a WSN node, this command can reach the coordinator node at any time of the working cycle. If the destination end device node is into active status, the coordinator node will send the message to the destination end device node inmediatly. But if the destination end device node is in sleep state, the coordinator node will wait until the destination end device node wakes up for sending its message. As a result, the response time will vary between a minimum value which depends on the response speed of the ZigBee protocol (of the order of 100 milliseconds), and a maximum value depending on the values given to the ST, SN and SP parameters.

In order to know the response time and the total latency, the user sends any command to any node, and when the response returns to the user interface, the web interface shows the response time of the WSN. To perform thisexperiment the DB command was used. This command requests the RSSI indicator value (Received Signal Strength). Table 2 shows the time measurements for different configurations of ST, SP and SN parameters. To perform the experiment, an end device node and the coordinator node were used. A series of 30 measurements for each combination of parameters was performed. The table shows the average value, the smallest value, the largest value and standard deviation for each set of 30 measurements for each combination of parameters.

Table 2: WSN response time

Parameters Value			**WSN response time**			
ST	**SP**	**SN**	**Average value**	**Shortest value**	**Longest value**	**Standar deviation**
30	30	1	827 ms	465 ms	1.30 s	244 ms
3E8	30	1	667 ms	349 ms	869 ms	227 ms
3E8	3E8	1	4.43 s	776 ms	10.2 s	3.27 s
3E8	3E8	3	10.1 s	630 ms	20.3 s	8.10 s

It can be noted that, for each combination of parameters, the shortest response time does not dependent on the parameter values, while the highest value, the average and standard deviation of the response times depend on the values given to ST, SP and SN. Rows 1 and 2 show that if the ST time interval increases, the average response time decreases. This is due to by increasing the time interval during which the destination node stays active, increases the probability that the request arrives to the WSN when the destination node is listening for new messages, and therefore it can respond immediately. On the other hand, by increasing the SP time interval or the SN parameter, the shortest response time does not increase, but the average response time, the longest response time and the standard deviation increase.

In order to perform this experiment when a remote laboratory is not accesible, the user must: (1) deploy a WSN and configure nodes with proper firmware according to their function, (2) write a program to communicate with the coordinator node through the USB port, (3) send data frames or commands to the nodes to perform the desired operations, and

(4) interpret the frames received by the coordinator to analyze the results. Therefore, the remote laboratory, in addition to allow the user to experiment with real hardware, eases the execution of the experiments. The user does not need to deploy a WSN, configure nodes from scratch, nor write programs for communicate with nodes. The user only need to select the desired values for the parameters and interpret the results.

5.2 Received RF Power

The RF power received by a node depends on the RF power transmitted by the transmitter node and the attenuation of the RF signal. The attenuation depends on many factors, including the distance between nodes, the presence of objects, the height of the nodes with respect to the ground plane, the position of antennas, etc. Most of commercially available nodes provide a mechanism to measure the received RF signal power. This allows programmers to adjust the transmission power to the required value.

The XBee nodes enable to know the RSSI value (indicator signal strength received) indicating the RF power received by a node in dBm. This RSSI value corresponds to the last data packet received by the node (which may be a data packet, a response, etc.). In addition, two parameters allow to adjust the transmitted power and receiver sensitivity. On the one hand, the PL command (Power Level) enable to adjust the transmit power of a node allowing choose from 5 values, as shown in the table 3. On the other hand, when the PM parameter is equal to 1, it increases the transmission power by 2 dB compared to the values shown in the table 3 and the receiver sensitivity by 1 dB.

With this experiment, the user can observe the variation of the RF power received by a node as a function of the RF power transmitted bythe sending node and the distance between nodes.

To perform this experiment, the user must

Table 3: Power transmitted by a node in functión of the PL parameter (according to the node XBee datasheet [22])

PL value	Output RF power
0	-7 dBm
1	-3 dBm
2	-1 dBm
3	+1 dBm
4	+3 dBm

use the DB command. This command requests the value of the RSSI indicator (received signal strength) of the last data packet received by the node in dBm [22]. The methodology used in this experiment is to modify the transmission power of the coordinator node, and verify the RSSI value of end device nodes. The node 1 is the closest to the coordinator node, then node 2, and so on. Table 4 shows the results of this experiment with some WSN nodes that form the remote laboratory. The RSSI values are expressed in dBm (as is explained in the XBee datasheet [22]). It can be noted that, the greater the transmission power or the shorter the distance between the nodes, the greater the RF power received by the destination node.

The realization of this experiment requires thatusers to know about configuration of WSN and about RF signals transmission. But users do not need to deploy a WSN, program theirnodes, or interpret data frames, because these tasks are performed by the remote laboratory.

5.3 Response Time as a Function of the Communication Channel Occupation

The XBee nodes used in this remote laboratory implement their physical layer and MAC sub-layer according to the IEEE 802.15.4 standard [23]. This uses the CSMA/CA protocol to access and share the wireless communicationchannel. According to the established by the IEEE 802.15.4 standard, when a node needs totransmit data, first verifies three times if thecommunication channel is free, adding random time intervals

Table 4: RSSI received by end device nodes to different RF powers of the coordinator node and different distances (in dBm)

Coordinator config.		RSSI received by end device nodes			
PM	PL	Node 1	Node 2	Node 6	Node 7
00	00	-30 dBm	-43 dBm	-67 dBm	-89 dBm
01	00	-28 dBm	-37 dBm	-61 dBm	-85 dBm
00	01	-28 dBm	-34 dBm	-61 dBm	-84 dBm
01	01	-27 dBm	-31 dBm	-57 dBm	-76 dBm

between each verification. If the communication channel is free for the three verifications, the node transmits its data. If the communication channel is busy in any of the three verifications, the node waits a random time and then restarts the verification process. This mechanism is intended to reduce the collisions probability.

The greater the number of nodes or greater data transmission frequency, the greater the probability that a node finds the communication channel busy and has to wait for sending data. As a result, the average response time increases. This experiment aims to measure the avegare response time of a node as a function of the transmission frequency of the other nodes in the network. In order to perform the experiment, the node 1 was programmed with a response time of approximately 2 seconds. The other 7 nodes (from node 2 to node 8) were configured to transmit data periodically. The response time of node 1 was measured, for different transmission frecuencies of the other seven nodes.

Nodes 2 to 8 are configured to transmit data periodically every 0.1 seconds, 0.5 seconds, 5 seconds, 50 seconds and never. For each one of these periods of data transmission, 30 measurements of node 1 response time were taken. Average value and standard deviation were calculated. Table 5 shows the results.

It can be noted that, the shorter the period of sending data of interfering nodes (and therefore greater the communication channel occupation), the longer the average response time of a node to a command. It can be noted

Table 5: Average response time for node 1 (in seconds) depending on the communication channel occupation

Period of sending data for nodes 2 to 8	Average response time		Standard deviation
0.1 s	9.60 s		9.69 s
0.5 s	8.19 s		7.10 s
5 s	2.54 s		1.93 s
50 s	2.07 s		0.75 s
nunca	1.99 s		0.46 s

that when nodes from 2 to 7 transmit data every 100 milliseconds or 500 milliseconds, the average response time and its standard deviation increase considerably. However, for period of sending data for nodes 2 to 7 longer than 5 seconds, the average response time of the node 1 does not vary considerably and is close to 2 seconds. This result can be attributed to that, the higher the communications channel occupation, the greater the probability that a node finds the communications channel bussy and it must wait to transmit data.

6 Conclusions and Future Work

This paper presents a remote laboratory aimed at user training deployed in a Cloud Computing infrastructure. A prototype has been built. Some proof of concept experiments have been designed, performed and analyzed. These proof of concept experiments can be performed quickly, since the user does not need to deploy a WSN nor a data acquisition system. Users can put all their attention on the concept that need to learn or train, without to waste time on configuring the experiment.

The proof of concept experiments presented in this paper are not trivial experiments. They test fundamental concepts about WSN that are important in real applications, for example:

- Time response of nodes WSN under different working conditions
- Received signal strength in function of distance and transmission power
- Congestion under different working conditions

This demonstrates that remote laboratories can be a powerful tool for training activities, useful for beginner or advanced users or students.

Future work includes:

- Do performance experiments, to know the limitations and shortcomings of the prototype, and to advance toward a final product
- Test the remote laboratory with students. This will allow to know their opinions, and to improve the design of the remote laboratory
- Test the remote laboratory with other types of system under test

References

[1] M. Waldrop, "Campus 2.0," Nature, vol. 495, no. 7440, pp. 160–163, 2013.

[2] M. M. Waldrop, "Education online: the virtual lab." Nature, vol. 499, no. 7458, pp. 268–270, 2013.

[3] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," Computer Networks, vol. 52, no. 12, pp. 2292 – 2330, 2008.

[4] E. Gilbert, "Research issues in wireless sensor network applications: a survey," International Journal of Information and Electronics Engineering, vol. 2, no. 5, pp. 702–706, 11 2012.

[5] G. Zhao, "Wireless sensor networks for industrial process monitoring and control: A survey," Network Protocols and Algorithms, vol. 3, no. 1, pp. 46–63, 2011.

[6] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation Computer Systems, vol. 25, no. 6, pp. 599 – 616, 2009.

[7] P. Mell and T. Grance, "The NIST definition of Cloud Computing (Draft)," NIST Special Publication, vol. 800, p. 145, 2011.

[8] A. Maiti and B. Tripathy, "Remote laboratories: Design of experiments and their web implementation." Journal of Educational Technology & Society, vol. 16, no. 3, 2013.

[9] L. Steyn and G. Hancke, "A survey of wireless sensor network testbeds," in AFRICON, 2011, Sept 2011, pp. 1–6.

[10] K. El-Darymli and M. Ahmed, Wireless Sensor Networks and Energy Efficiency: Protocols, Routing and Management. IGI Global, 2012, ch. Wireless Sensor Network Testbeds, pp. 148–205.

[11] 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.

[12] P. Godoy, L. Iacono, R. Cayssials, and C. García Garino, "A survey of WSN testbeds deployment," Congreso Argentino de Sistemas Embebidos (CASE), 2012.

[13] P. Orduna, A. Gomez-Goiri, L. Rodriguez-Gil, J. Diego, D. Lopez-de Ipina, and J. Garcia-Zubia, "wCloud: Automatic generation of WebLab-Deusto deployments in the Cloud," in Remote Engineering and Virtual Instrumentation (REV), 2015 12th International Conference on, Feb 2015, pp. 223–229.

[14] M. Tawfik, C. Salzmann, D. Gillet, D. Lowe, H. Saliah-Hassane, E. Sancristobal, and M. Castro, "Laboratory as a service (LaaS): A model for developing and implementing remote laboratories as modular components," in Remote Engineering and Virtual Instrumentation (REV), 2014 11th International Conference on, Feb 2014, pp. 11–20, 13 citas.

[15] S. Misra, S. Chatterjee, and M. Obaidat, "On theoretical modeling of sensor Cloud: A paradigm shift from wireless sensor network," Systems Journal, IEEE, vol. PP, no. 99, pp. 1–10, 2014.

[16] L. Iacono, C. García Garino, O. Marianetti, and C. Párraga, "Wireless sensor networks: A software as a service approach," Prospective and Ongoing Projects, VI Latin American Symposium on High Performance Computing (HPCLatAm 2013), Mendoza, Argentina, pp. 184–195, 2013.

[17] L. Iacono, P. Godoy, O. Marianetti, C. García Garino, and C. Párraga, "Estudio de la integración entre WSN y redes TCP/IP," Memoria de Trabajos de Difusión Científica y Técnica 10, 2012, ISSN 1510-7450.

[18] 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.

[19] O. Marianetti, Laboratorios remotos, un aporte para su diseño y gestión. Universidad de Mendoza, Facultad de Ingeniería, Maestría en Teleinformática, 2006.

[20] K. Jona and D. Uttal, "Don't forget the teacher: New tools to support broader adoption of remote labs," in Remote Engineering and Virtual Instrumentation (REV), 2013 10th International Conference on, Feb 2013, pp. 1–2.

[21] Google Inc. (2013) Google App Engine website. Retrieved in 2015. [Online]. Available: https://appengine.google.com/

[22] Digi International Inc. (2008) XBee ZB low power ZigBee module with integrated wire antenna datasheet. Retrieved in 2015. [Online]. Available: http://www.digi.com/products/model?mid=3008

[23] Institute of Electrical and Electronic Engineers, "IEEE std 802.15.4-2003," pp. 1–670, 2003.