HPCLatAm 2013, pp. 184-195 (full paper)
Session: Prospective and Ongoing Projects
C. Garcia Garino and M. Printista (Eds.)
Mendoza, Argentina, July 29-30, 2013

Wireless Sensor Networks: A Software as a Service Approach

Lucas Iacono[1,2], Carlos García Garino[2,3], Osvaldo Marianetti[1], and Cristina Párraga[4]

[1] Instituto de Microelectrónica. Facultad de Ingeniería, Universidad de Mendoza.
Mendoza, Argentina.
`{lucas.iacono,osvaldo.marianetti}@um.edu.ar`
[2] ITIC, Universidad Nacional de Cuyo. Mendoza, Argentina.
[3] Facultad de Ingeniería, Universidad Nacional de Cuyo. Mendoza, Argentina.
`cgarcia@itu.uncu.edu.ar`
[4] DICYTyV, Universidad de Mendoza. Mendoza, Argentina.
`cristina.parraga@um.edu.ar`

Abstract. This paper presents a new integration system to achieve the WSNs remote access through Cloud Computing Services. The proposed system provides easy and reliable remote access to data and settings of agro-meteorological WSNs. In order to implement the proposed system experiments with two Cloud Computing Services, Globus Online and Google Drive were conducted. In the paper different experiments are discussed in order to show the proposed system capabilities. The tests shown that agro-meteorological WSNs, composed by nodes with limited resources, can be easy accessed through Cloud Computing Services.

1 Introduction

Agricultural monitoring systems are a useful tool in order to ensure production quality, traceability and prevent crop damage due to phenomena like frost. An agricultural monitoring system involves two main aspects: the in-field data acquisition and the access to the collected data for its further analysis.

The in-field data acquisition (DAQ) can be made using measuring instruments, weather stations and Wireless Sensor Networks (WSNs). Compared with instruments and weather stations WSNs have the following advantages for agricultural monitoring: low maintenance, reliability, robustness, and scalability [1].

Remote access to data collected by DAQ tools like WSNs can be done using ad-hoc applications or web servers. While these technologies allow access from Internet to WSNs data and setup, present the follow problems: (i) generally require that users have knowledge of WSN programming, (ii) they are not prepared to scale if will increase the volume of data due to new WSNs connected to the system and (iii) do not provide fault tolerance and high reliability.

These problems can be solved using Cloud Computing technologies. The use of Cloud Computing services allows the develop of agricultural monitoring systems that takes advantage of this technology (data replication, fault tolerance, resources scaling, etc.). However, the manage of WSNs data and settings through

Cloud Computing presents two main problems to solve: the incompatibility between WSNs and Cloud protocols, and the integration of WSNs with limited hardware and energy resources to Cloud Computing.

This paper discusses a new agricultural monitoring system based in Cloud Computing Software provided as a Service (SaaS). The proposed system will provide remote and simple access to WSNs data and settings. The main goals of the proposed system will be: (i) to keep the well known properties of WSN like low energy and hardware resources consumption; (ii) to use technologies available at local market. Background of the proposed system can be found in references [2,3].

The rest of the work is organized as follows: in Section 2 an introduction to WSNs is presented. Next, Section 3 gives an overview of Cloud Computing technologies. Section 4, presents a discussion of the interconnection of WSNs to TCP/IP. Section 5 details the WSN SaaS system architecture and description. Then, Section 6 includes the WSN SaaS Prototype implementation and Section 7 the conducted experiments. Finally, in Section 8 conclusions are provided.

2 Wireless Sensor Networks

The present Section gives an introduction to WSNs including hardware, applications, characteristics and protocols.

2.1 Sensor Nodes

In the Late 90s, a new kind of embedded systems arises. These devices called sensor nodes are integrated by a micro-controller, memory, different sensors, battery and a RF transceiver.

Sensor nodes can be interconnected forming Wireless Sensor Networks (See Figure 1) and interact between them. Such networks are used for the study of the environment and to acquire different variables (temperature, humidity, pressure, etc.).

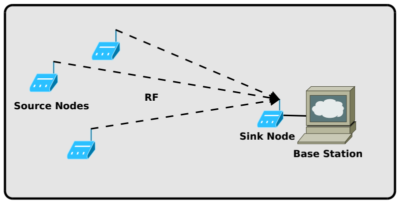

Fig. 1. Wireless Sensor Network.

In a WSN, data are acquired by nodes called sources and sent via RF to a special node called base station. The base station coordinates the operation of the WSN and can be a personal computer (PC) or an embedded system. Furthermore, the base station can store or transmit via Internet all the information registered by the network. Nodes must meet requirements such as autonomy, low power consumption, low cost, robustness and reliability.

2.2 WSNs Protocols

Unlike traditional wireless networks, WSNs are composed of nodes with limited hardware and energy resources. Therefore nodes must use communications protocols specifically designed to work with scarce energy sources and hardware resources. In addition, these protocols are not compatible with the TCP / IP stack protocols.

There are different protocols of Physical layer and MAC sub-layer for wireless sensor networks. However, the IEEE 802.15.4 standard [4] has emerged as one of the most popular in the field of WSNs. The popularity of IEEE 802.15.4 is due to it is simple to implement and fully complies with the performance requirements of WSNs. Furthermore, in recent years the number of hardware manufacturers that provide IEEE 802.15.4 has increased. The main applications of IEEE 802.15.4 include WSNs, home automation, health monitoring, etc.

Currently there are different protocols that add functionalities to IEEE 802.15.4 like ZigBee. ZigBee adds the network layer, security services and application support sublayer (APS) above the IEEE 802.15.4 MAC sublayer. These features added by ZigBee allow mesh network topologies, security and application profiles (health, home automation, etc.).

These protocols have been widely discussed by different authors [5,6,7] therefore are not going to be analyzed in this paper.

3 Cloud Computing

Cloud Computing is a new paradigm for application development and the use of computing and storage resources [8]. Through the use of virtualization techniques and web services, hardware resources and applications can be dynamically provided to the user.

Mell and Grance [9] define Cloud Computing as “a model for enabling ubiquitous, convenient, on demand network access to a shared pool of configurable computing resources (i.e. networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This Cloud model is composed of five essential characteristics, three services models (Software / Platform / Infrastructure as a Service), and four deployment models, whereas the five characteristics are: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured services. The deployment models include private, community, public and hybrid Clouds”.

One of the main advantages of Cloud Computing is the scalability of resources. Through scalability, Cloud Computing can solve the computational and storage requirements of the applications. Another advantage of Cloud Computing is that the users can easily access to development frameworks of applications that use Cloud Services in order to allow the scaling of resources.

Cloud Computing offers different kind of services. These Cloud services can be grouped in: Infrastructure Services, Platform Services and Application Services.

Infrastructure as a Service (IaaS) Through infrastructure services the users can access to virtualized high performance computing resources (CPUs, storage devices, etc.). The service provider delivers resources to the client in accordance to the specific requirements requested: type and CPU power, memory, storage, operating system, etc.

Platform as a Service (PaaS) The Platform services provide Application Programming Interfaces (APIs) and standard development kits (SDKs) in order to allow users to develop and implement their own applications for Cloud Computing. Some of these platforms are Google App Engine [10] and Amazon EC2 [11].

Software as a Service (SaaS) These services are applications that can be accessed by an end user through a Internet connection and a standard web browser. Furthermore, the applications can be developed with Platform Services and executed with Infrastructure Services. As an example of SaaS we can cite Google Drive [12] and Globus Online [13].

4 WSN - TCP/IP Integration

In order to solve the integration between WSNs and TCP/IP a properly communication architecture is required. Currently there are two main approaches that solve the communication architecture: Gateway and Overlay Networks [14,15].

Gateway. In a Gateway architecture the base station acts as a gateway translating both networks protocols: TCP/IP and e.g. ZigBee [14]. The Gateway has the physical interfaces and protocols of both networks. Furthermore, the base station is the unique point of access to the WSN. This device allows to take the data and commands of WSNs, in the native protocol, and share them with TCP / IP. Then, through a larger network such as the Internet, are delivered to the application in order to allow the user access.

Overlay Networks. This approach allows compatibility between two networks that run under different protocols by overlapping one of the protocols on the other. In the special case of WSN - TCP/IP integration, the Overlay approach is called TCP/IP "Overlay" Sensor Networks because a full or special version of the TCP/IP stack is embedded on the sensor nodes [14].

The WSN - TCP/IP interconnection have been analyzed in detail in previous works made by the authors [2,3]. These studies shows that Gateway approach consumes fewer processing and energy source resources in WSNs nodes that

Overlay Approach. Therefore, Gateway approach will be used in the proposed system for interconnect agro-meteorological WSNs to TCP/IP.

5 WSNs: Software as a Service

Cloud Computing Services are used in order to export the data and configuration of the WSNs to the users. Previous works [16,17,18,19,3] show that Clouds Services in sensor clouds can be embedded or not in WSN elements. The main advantage of not embedded approach is a lower energy consumption in WSN elements, consequently it is chosen in this work.

5.1 Architecture

The architecture of the proposed integration system is shown in Figure 2. It is based in two main modules and a web application. The first software module is the WSN-TCP/IP interconnection one described in Section 5.2. The Cloud service module is designed in order to implement a data transfer service for the WSN. Both data and management commands are sent in this way through the WSN-TCP/IP interconnection module.

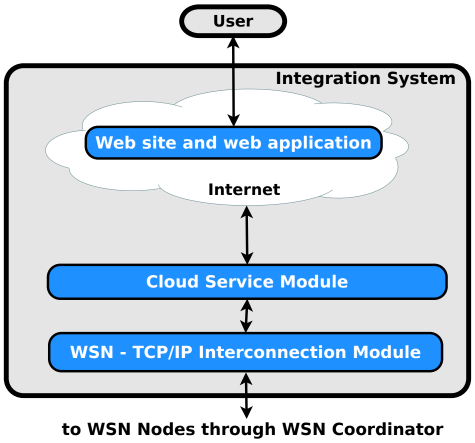

Fig. 2. WSN SaaS Integration System Architecture Overview.

The website provides access to a web application that allows to the users to interact with the Cloud Service module in order to manage WSN data and configuration files. In addition the web appi process the WSN data and detect possible frost cases. In Section 5.2 these modules are described and its implementation is discussed in Section 6.

5.2 Software Modules Description

Figure 3, shows WSN - TCP/IP Interconnection module, through which the system receives the WSN data and sends it to the Cloud Service Module. The process begins when the ZigBee Coordinator receives the data from WSN nodes (1). After reception, the ZigBee Coordinator sends the data (2) to the base station (a PC or embedded system). The ZigBee Coordinator is connected to the base station via USB or RS232. In the proposed system the base station hosts the WSN - TCP/IP Interconnection Module of the system.

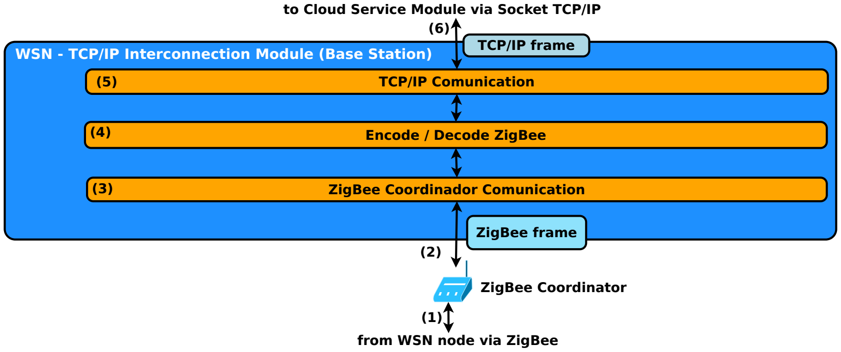

Fig. 3. WSN - TCP/IP Interconnection Module.

WSN - TCP/IP Interconnection Module (See Figure 3) includes different processes that enable communication with ZigBee Coordinator (3) in order to decode and to extract data from ZigBee frames (4). Finally the extracted data are encapsulate into TCP/IP frames (5). Once encapsulated, the data are sent to a remote machine which contain the Cloud Service Module. Communication between the two modules is done through a socket TCP/IP over 802.11 or Ethernet (6).

Figure 4 shows the Cloud Service Module. In this module the Integration Module receives frames from the base station (7). Then, data are managed by the Data Access Module. This module has two main processes. The first one is called "Storage Process" and generates a database file with WSN measurements (8). Next the second process, called "Access Process" (9), allows to the Cloud Service the handling of the database files in order to export them to Internet (10).

Finally, the user can access to WSN data through a website that includes a web application. The web application allows to the user sends to the Cloud Service Module the data access requests.

Similarly to the data acquisition process, the user can change the configuration of the nodes via the Internet. As an example, the change in sampling frequency of WSN nodes is detailed.

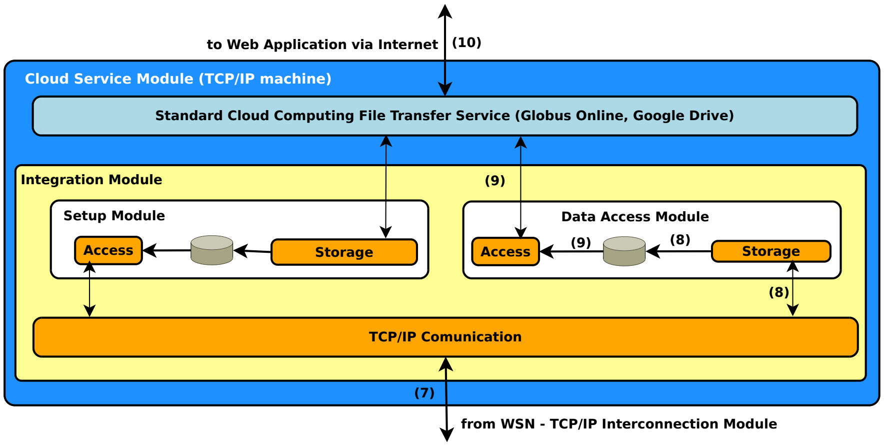

Fig. 4. Cloud Service Module.

First, the user accesses the website and sends the change request to the Cloud Service Module through a web application. Then, in Cloud Service Module the user request is received via Cloud File Transfer Service and the Integration Module requests to the Setup Module to store the new sampling frequency value locally into a file that in turns is sent to the WSN - TCP/IP Interconnection Module. In this module the sampling frequency value is encoded in a ZigBee frame and sent to the WSN Coordinator. Finally the transceiver of the node receives the new frequency value and performs the requested change.

6 Prototype Implementation

The present Section details the WSN SaaS prototype implementation. Currently an experimental WSN and the WSN-TCP/IP Interconnection Module are fully implemented. A partially implemented but functional Cloud Service is in progress.

6.1 Experimental WSN

The experimental WSN is based in nodes integrated by a Atmega micro-controller in a Arduino Pro Platform, Sensirion SHT15 temperature-humidity sensor, XBee ZigBee Series 2 Transceiver, IP65 waterproof case and batteries. This WSN will be used in order to study frosts in fruit trees.

6.2 WSN - TCP/IP Interconnection Module

The WSN - TCP/IP interconnection module was programmed in the base station using Python 2.7 over a Linux Ubuntu 12.04. The developed software include a serial connection to the WSN coordinator and the ZigBee encode-decode processes. In addition the base station acts as a web client that sends the data, through the socket library of Python, to a server (the Cloud Service host machine) via TCP/IP socket.

6.3 Cloud Service Module

A software module receives the WSN data from the WSN - TCP/IP Interconnection Module in order to storage it in a database. Next, a Standard Cloud File Transfer Service was used to export the data base files to the web application via Internet.

There are different Standard Cloud File Transfer Services that can be used for export the WSNs resources. However two potential candidates were selected, one from academic field (Globus Online) and other from enterprise side (Google Drive - App Engine).

Globus Online (GO) [13], was developed by the Computation Institute of the University of Chicago and the Argonne National Laboratory. GO is aimed to provide SaaS that facilitate the transfer of large volumes of data between users.

The GO transfer services use two types of nodes: source and sink, that denote the machines that send and receive the data respectively. In GO both nodes are called "endpoints". Thanks to the use of SaaS, the user only has to install a software module (Globus Connect) and not the complete Globus Middleware in both endpoints.

Currently GO does not allow the direct transfer between endpoints. Therefore, the process of data transfer requires that one side of a transfer must be a GridFTP server with a globally routable IP address. The GridFTP server can belong to the user or can be a public GridFTP server of the Globus Online Cloud. However, if the client use a GO public GridFTP server, the size of files to transfer will be restricted to 10MB.

Google Drive [12], is a service that allows upload, view, edit, sync and share files. The user can access to Google Drive services either on-line (through a standard Web browser) or locally via Google Client. Although Google Drive Client provides all Google Drive services, currently is only available for Mac OS, Windows and Android.

The free hosting space of Google Drive is 5 GB. Furthermore, it has an API that supports different programming languages (Python, Java, etc). Google Drive API enables to the developers include in their applications the different services of Google Drive.

In addition Google provides Google App Engine [10], a tool that allows the development of applications involving Google Drive services. Google App Engine includes development kits for different languages (Python, Java, etc.). Through this tool, the developer can upload their applications and run them on Google

Cloud servers using an appropriate execution environment (Python interpreter, JAVA virtual machine, etc.).

Google App Engine allows free hosting of up to ten applications in Google Cloud. Furthermore, it also provides scaling services that increase the computing power according to application requirements.

7 Experiments

The present Section includes the different experiments conducted with the developed prototype, using GO and Google Drive.

7.1 Experimental WSN

The in-field DAQ process has been conducted with the experimental WSN proposed in this work. First tests of connectivity between nodes and the base station has been conducted both in controlled laboratory and outdoor conditions [1]. Next the WSN was deployed in a fruit farm in General Alvear, Mendoza in order to perform in field DAQ of agro-meteorological variables (temperature and humidity).

7.2 WSNs SaaS

The experiments conducted with WSN - TCP/IP Interconnection Module allowed to extract in the base station the temperature and humidity data from the ZigBee frames and via TCP/IP Socket send them to a remote computer. These WSN data pre-processing allows to send data via socket TCP/IP without the overhead of ZigBee frame headers.

The bandwidth required for base station is 64 bytes/sec. because the base station transmits a frame of 32 bytes every 0,5 seconds. These delay (0,5 sec.) is required in order to pre-process the frame in the base station.

GO based tests were carried out on two PCs running Ubuntu 12.04. In the first machine it was implemented, with Python 2.7, a software module that receive the data from the base station and storage them into data base files. Next a SSH public key was generated in both machines. Then an account of GO on the domain www.globusonline.org was created. In turn the Globus Online application was installed in both end points and were registered in Globus Cloud through the GO web interface. Finally the WSN data base files was transfered using a public GridFTP server of Globus Online Cloud. The Figure 5 illustrates the WSNs data base files transfer process performed through Globus Online.

The experiences with Google Drive (See Figure 6) were conducted on a Windows 7 PC connected to the WSN base station through socket TCP/IP. In the PC we host the software module that allow to store in data base files the WSNs data and a Google Drive Client. That service was used to export the WSNs data files to Internet. Then the files were accessed from Google Drive Website with a Google Account and from a Android Smartphone with Google Client.

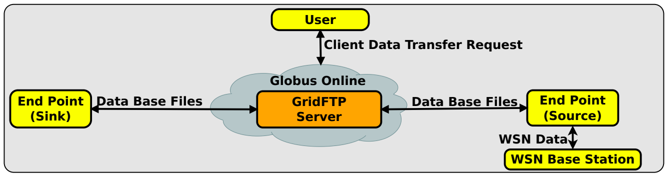

Fig. 5. WSN Data Transfer Process with Globus Online.

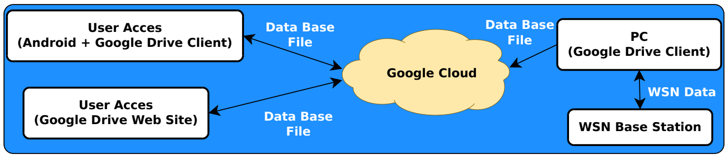

Fig. 6. WSN Data Transfer Process with Google Drive.

The storage space of GO and Google Drive is enough to perform the transfer of WSN data and setup files. As an example it can cite that in a extreme operating condition of the system (10 WSNs with 10 nodes per network and each one transmitting a data per second) the data volume will be 255 MBytes per day.

While these volume of data exceeds the storage space of the example servers of GO and in one month the basic storage of Google Drive, both Cloud Services offers different solutions.

In the case of GO it can deploy a public gridFTP server that stores these volume of data. On the other hand Google Drive has paid services that allows more storage space. In addition, historical WSNs data can be downloaded to a local server leaving in the Cloud only the recent data.

Google Drive provides APIs and SDKs with more services than GO. Furthermore GO API is still in beta testing. Google Drive allows that its services can be integrated into web applications with scalable computing resources provided by Google Cloud. In addition Google Drive provides compatibility with Android mobile devices allowing access from mobile platforms to WSNs data and configuration.

In terms of WSN SaaS system scalability, GO is designed to enable reliable transfer of large volumes of data. Then GO would be more appropriate than Google Drive to implement a WSN SaaS integration system designed to manage WSNs composed of large number of nodes.

GO is endorsed by the Globus Alliance. This academic community offers support to developers and researchers. Besides, GO is a free and open service which enables access to it source code GO.

In practice both alternatives are possible, when large data files need to be transfered perhaps GO is more suitable. On the other hand GO is a free cost application and source code is available. However Google Drive has flexible tools and can be useful for many applications.

Finally it can says that preliminary test of the system has been carried out successfully. According expected transmission rates (64Bytes/sec.) the system operation is correct. However, is ongoing a more severe in-field test in order to study transmission rates, power consumption in the WSN nodes and delays.

8 Conclusions

In this work a SaaS prototype designed in order to provide WSN remote access and management has been discussed. The proposed tool solves interconnection of WSNs protocol with TCP/IP ones. On the other hand a Cloud Service Module lets the access to WSN data and settings in a transparent way.

Two different Cloud Services have been tested Globus Online and Google Drive. Google Drive have a nice API and provide more services than GO. However, GO comes from academic field and perhaps can be attractive to have access to source code and for large data files transfer. In this work Google Drive has been chosen to develop the Cloud Service Module.

The proposed WSN SaaS was tested. First in field DAQ of temperature and humidity have been conducted. Then, the obtained data was sent to a remote computer and stored in proper data base.

Finally the discussed WSN SaaS provides to scientist and farmers easy access to remote WSN. Therefore the region would have a new tool that allows the study of agro-meteorological phenomena (frost, storms, etc.) in order to avoid economic losses.

Acknowledgements

The authors acknowledge the financial support provided by the project PAE-PID 146 granted by Argentinean Agency for R&D activities (ANPCyT) and Instituto de Desarrollo Industrial Tecnológico y de Servicios (IDITS) and project PAE-PICT 2312 granted by ANPCyT. The first author acknowledges his type II doctoral scholarship provided by CONICET.

References

1. P. Godoy, L. Iacono, R. Cayssials, C. Párraga, C. García Garino, Effect of working conditions over the performance in ZigBee WSN, in: Capítulo Sistemas de Control, Anales de la primera reedición del Congreso de la Sección Argentina del IEEE (Argencon 2012)., Córdoba, Argentina, 2012, ISBN: 987-572-076-3.
2. L. Iacono, P. Godoy, O. Marianetti, C. García Garino, C. Párraga, Estudio de la Integración entre WSN y redes TCP/IP, Memoria de Trabajos de Difusión Científica y Técnica 10, ISSN 1510-7450.
3. L. Iacono, Acceso Remoto a Redes de Sensores Inalámbricas Mediante Tecnologías de Computación Distribuída, Thesis Proposal, Facultad de Ingeniería, Universidad de Mendoza (2013).

4. IEEE, IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks-Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003) (2006) 1–305doi:10.1109/IEEESTD.2006.232110.
5. P. Baronti, P. Pillai, V. Chook, S. Chessa, A. Gotta, Y. Hu, Wireless Sensor Networks: A Survey on the State of the Art and the 802.15. 4 and ZigBee Standards, Computer Communications 30 (7) (2007) 1655–1695.
6. S. Ergen, ZigBee/IEEE 802.15. 4 Summary, Available in: http://pages.cs.wisc.edu/ (2004).
7. P. Kinney, Zigbee Technology: Wireless Control that Simply Works, in: Communications design conference, Vol. 2, 2003.
8. R. Buyya, C. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility, Future Generation Computer Systems 25 (6) (2009) 599–616.
9. P. Mell, T. Grance, The NIST Definition of Cloud Computing, Recommendations of the National Institute of Standards and Technology, NIST Special Publication (2011) 800–145.
10. Google, Google App Engine, Available in: https://developers.google.com/appengine/ (February 2013).
11. Amazon EC2, Amazon Web Services, Available in: http://aws.amazon.com/es/ec2/ (November 2012).
12. Google, Google Docs, Available in: https://drive.google.com/ (February 2013).
13. I. Foster, Globus Online: Accelerating and Democratizing Science Through Cloud-Based Services, Internet Computing, IEEE 15 (3) (2011) 70–73.
14. R. Roman, J. Lopez, Integrating Wireless Sensor Networks and the Internet: a Security Analysis, Internet Research 19 (2) (2009) 246–259.
15. M. Zuñiga, B. Krishnamachari, Integrating Future Large-Scale Wireless Sensor Networks with the Internet, USC Computer Science Technical Report CS 03-792, Available in: ftp://ftp.usc.edu/pub/csinfo/tech-reports/papers/03-792.pdf (2003).
16. M. Yuriyama, T. Kushida, Sensor-Cloud Infrastructure-Physical Sensor Management with Virtualized Sensors on Cloud Computing, in: 13th International Conference on Network-Based Information Systems (NBiS), IEEE, 2010, pp. 1–8.
17. M. Yuriyama, T. Kushida, M. Itakura, A New Model of Accelerating Service Innovation with Sensor-Cloud Infrastructure, in: 2011 Annual SRII Global Conference (SRII), 2011, pp. 308 –314. doi:10.1109/SRII.2011.42.
18. K. Lee, D. Hughes, System Architecture Directions for Tangible Cloud Computing, in: International Workshop on Information Security and Applications (IWISA 2010), Vol. 25, Qinhuangdao, China, 2010.
19. K. Lee, D. Murray, D. Hughes, W. Joosen, Extending Sensor Networks Into the Cloud Using Amazon Web Services, in: 2010 IEEE International Conference on Networked Embedded Systems for Enterprise Applications (NESEA), IEEE, 2010, pp. 1–7.