



UNIVERSIDADE DA BEIRA INTERIOR
Faculdade de Engenharia
Departamento de Informática

Ubiquitous Model for Wireless Sensor Networks Monitoring

André Gaudêncio Ferreira Elias

Submitted to the University of Beira Interior in candidature for the
Degree of Master of Science in Informatics Engineering

Supervised by Prof. Doutor Joel José Puga Coelho Rodrigues
Co-supervised by Prof. Doutor Bruno Bogaz Zarpelão

Department of Informatics
University of Beira Interior
Covilhã, Portugal
<http://www.di.ubi.pt>

Acknowledgments

First of all, I would like to express my gratitude to my supervisor Professor Joel José Puga Coelho Rodrigues for his expertise, guidance, continuous support and words of encouragement.

I also want to thank to my co-supervisor Professor Bruno Bogaz Zarpelão for all the support throughout this work and for the fantastic reception and guidance during my Master's studies at University of Campinas (UNICAMP), in Brazil.

I am most grateful to the University of Beira Interior, *Instituto de Telecomunicações* (Covilhã Delegation), Next Generation Networks and Applications Group (NetGNA), and University of Campinas (UNICAMP) for many kinds of all the support that was given to me.

I owe particular thanks to Professor Luís Oliveira and my colleague Gilberto Almeida, who worked closest to me and gave me all the necessary support during this work.

Many thanks to all members of NetGNA for the constant support and for creating an excellent work environment.

Special thanks to my friends David Albuquerque and Patrícia Nunes for keeping me focused and motivated during this months of hard work and dedication.

Last but not least, I am most grateful to my family, especially my mother Maria Fernanda Gaudêncio Elias, my brother Bruno Elias and my cousins Pedro Gaudêncio, Susana Baptista, Carolina and Mafalda, for their support, love, patient and constant encouragement.

I dedicate this work to my father António Gabriel Elias for making me the person I am today.

Resumo

As redes de sensores sem fios fazem parte de uma nova tendência tecnológica na qual pequenos dispositivos com recursos limitados comunicam entre si, sem fios, e interagem com o ambiente envolvente recolhendo uma grande diversidade de dados, tais como a temperatura e a humidade.

Recentemente, devido ao enorme crescimento no uso de dispositivos móveis com ligação à Internet, os *smartphones* estão a tornar-se o centro das futuras redes sem fios ubíquas permitindo aos utilizadores aceder a dados, a qualquer hora e em qualquer lugar. De acordo com a visão da *Internet of Things*, interligar redes de sensores sem fios e *smartphones* usando a Internet é um grande desafio e novas arquitecturas são necessárias devido à heterogeneidade destes dispositivos.

Esta dissertação centra-se na proposta e construção de uma arquitectura ubíqua para a monitorização de redes de sensores sem fios, baseada em serviços Web, apoiada numa base de dados relacional e uma aplicação móvel para o sistema operativo Android. Esta arquitectura permite que os utilizadores móveis acessem a dados em tempo real e também a dados históricos, num ambiente móvel, usando *smartphones*. Além disso, foi desenvolvido um sistema de notificações *push* que alerta o utilizador quando um dado parâmetro de um sensor ultrapassa um limiar pré-definido.

A solução construída foi testada e demonstrada utilizando uma testbed laboratorial e está pronta para utilização.

Abstract

Wireless Sensor Networks (WSNs) belongs to a new technology trend where tiny and resource constrained devices are wirelessly interconnected and are able to interact with the surrounding environment by collecting data, such as temperature and humidity.

Recently, due to the huge growth of mobile devices usage with Internet connection, smartphones are becoming the center of future ubiquitous wireless networks allowing users to access data network services, anytime and anywhere. According to the Internet of Things vision, interconnecting WSNs with smartphones and the Internet is a big challenge. Then, due to the heterogeneity of these devices new architectures are required.

This dissertation focuses on the design and construction of a ubiquitous architecture for WSNs monitoring based on Web services, a relational database, and an Android mobile application. This architecture allows mobile users accessing real-time or historical data in a ubiquitous environment using smartphones. Besides that, a push notification system to alert mobile users when a sensor parameter overcomes a given threshold was created.

The entire solution was evaluated and demonstrated using a laboratory WSN testbed, and is ready for use.

Keywords

Wireless sensor networks; Ubiquitous computing; Internet of Things; IoT; Network monitoring; Environmental monitoring; Mobile computing; Web services; Mobile monitoring application; Push notifications; Real-time monitoring; Android.

Contents

Acknowledgments.....	iii
Resumo	v
Abstract	vii
Keywords	ix
Contents.....	xi
List Of Figures	xiii
List Of Tables	xv
Acronyms	xvii
1. Introduction	1
1.1. FOCUS.....	1
1.2. PROBLEM DEFINITION AND OBJECTIVES	2
1.3. MAIN CONTRIBUTIONS.....	3
1.4. DISSERTATION ORGANIZATION	4
REFERENCES.....	7
2. A Ubiquitous Model For Wireless Sensor Networks Monitoring.....	9
ABSTRACT	11
I. INTRODUCTION	11
II. RELATED WORK	12
III. MODEL ARCHITECTURE	12
IV. CONSTRUCTION OF THE PROPOSED MODEL.....	13
V. PERFORMANCE EVALUATION AND DEMONSTRATION	14

VI. CONCLUSIONS AND FUTURE WORK	15
REFERENCES.....	15
3. Ubiquitous Architecture For Wireless Sensor Networks Monitoring With Push Notifications And End-To-End Connectivity	17
ABSTRACT	19
KEYWORDS	20
1. INTRODUCTION	21
2. RELATED WORK	24
3. SYSTEM ARCHITECTURE	28
4. CONSTRUCTION OF THE PROPOSED MODEL	32
4.1 Database Design	32
4.2 Rest Web Service	34
4.3 Push Notifications	36
4.4 End-To-End Connectivity	38
5. ANDROID APPLICATION	39
5.1 Android User Interface.....	39
6. PERFORMANCE EVALUATION AND DEMONSTRATION	42
7. CONCLUSIONS	45
ACKNOWLEDGMENTS	46
REFERENCES.....	47
AUTHOR BIOGRAPHIES	49
4. Conclusions And Future Work.....	51

List of Figures

CHAPTER 1

Figure 1 - Illustration Of The System Architecture Diagram	13
Figure 2 - Photo Of The Used Laboratory Testbed And Gateway	14
Figure 3 - Login And Data Visualization Screens	15
Figure 4 - History And Settings Screens.....	15

CHAPTER 2

Figure 1 - System Architecture Diagram.....	28
Figure 2 - Database Entity-Relationship Diagram	33
Figure 3 - Restful Web Service Architecture.....	35
Figure 4 - Sequence Diagram Of The Push Notification System.....	37
Figure 5 - Login Screen.....	40
Figure 6 - Data Visualization Screen	40
Figure 7 - History Data Screen	42
Figure 8 - Settings Screen	42
Figure 9 - 6LoWPAN Wireless Sensor Network Laboratory Testbed	43

List of Tables

CHAPTER 2

Table I - Presentation of the Evaluation And Validation Tests.....	45
--	----

Acronyms

API	Application Programming Interface
CPU	Central Processing Unit
CSV	Comma Separated Values
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
LAN	Local Area Network
MAC	Media Access Control
MID	Mobile Internet Devices
OS	Operating System
PC	Personal Computer
PDA	Personal Digital Assistant
RAM	Random Access Memory
REST	Representational State Transfer
SDK	Software Development Kit
TCP	Transmission Control Protocol
XML	Extensible Markup Language
TinyOS	Tiny Operating System
UI	User Interface
URI	Uniform Resource Identifier
USB	Universal Serial Bus
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network

1. Introduction

1.1. Focus

Wireless sensors are small and autonomous devices capable for measuring all sorts of environmental and physical phenomena. In terms of hardware, sensor devices are resource constrained and each mote has several modules, such as a limited processing unit, an antenna for wireless communication, a low power battery, and the sensor modules attached to the circuit board. Based on these devices, a new emerging network technology known as wireless sensor networks (WSNs) has become a new trend in technological research [1, 2]. These networks may combine thousands of wireless sensor devices that are deployed in the surrounding environment allowing remote monitoring of physical and environmental variables, such as temperature, pollution, and natural disasters [3-5].

The main challenges regarding wireless sensor networks are power consumption and their connection to the Internet, as defended by the Internet of Things paradigm [6, 7]. The use of the TCP/IP stack in WSNs, as initially conceived, is too heavy for the resource-constrained devices. The standardization of IPv6 over Low Power Personal Area Networks (6LoWPAN) allows the transmission of IPv6 packets over the standard IEEE 802.15.4, enabling WSNs to communicate with the Internet more efficiently [8-11].

With the enormous growth of Internet-enabled personal mobile devices and operating systems, it is a big challenge to establish connectivity between personal mobile devices and wireless sensor networks allowing remote and ubiquitous monitoring of wireless sensor networks. The integration between wireless networks and mobile devices, specifically

smartphones, is crucial for the Internet growth. With the huge dissemination of smartphones, the growing diversity of platforms and mobile operating systems is developing heterogeneous network scenarios. Therefore, is essential to develop new architectures to interconnect all these sorts of devices in a platform independent way [12-14].

Wireless sensor networks are mainly used for monitoring purposes. In order to increase the efficiency and usability of these networks in a ubiquitous environment, the mobile user should be alerted when there are significant changes in sensed data. Therefore, the integration of push-based notifications in WSNs monitoring enables the mobile user to be alerted when he/she is not actively monitoring the network neither using the smartphone, resulting in significant energy savings [15].

This dissertation proposes a ubiquitous monitoring solution for WSNs. The mobile user can monitor the network using a smartphone with Internet connection, anytime, anywhere. The user can access real-time data or historical measures and even choose the update rate of the collected data on the mobile application. Besides that, the user is alerted by push notifications when a sensor reading overcomes a given threshold.

1.2. Problem Definition and Objectives

The problem addressed in this dissertation is the connection of wireless sensor networks and the Internet in order to allow mobile and ubiquitous monitoring of these networks. Therefore, new architectures are needed to enable multi-platform mobile devices to access WSN collected data in mobility environments.

This dissertation aims to demonstrate how mobile computing, specifically smartphones, can be used with WSNs to remotely monitor physical and environmental phenomena. The data collected by sensors could be stored and analyzed to study environmental patterns of the

monitored environment. Moreover, using end-to-end connectivity between smartphones and sensors, real-time monitoring can be performed, giving the system the ability to immediately notify users that significant changes occurred on the monitored environment.

In order to reach this main objective, the following intermediate objectives were defined:

- Extensive review of the state of the art related with WSNs, data collection solutions, and mobile computing;
- Requirements analysis and design of the proposed solution;
- Construction of a Web service to allow the access to the database with historical sensed data by the mobile device over the Internet;
- Design and construction of a relational database to store WSN data;
- Development of an Android OS application for data presentation and performance evaluation experiments;
- Construction and configuration of a push notification system to alert users when sensor readings overcome pre-defined thresholds.
- Construction and deployment of a Web service module in order to establish end-to-end connectivity, between mobile phones and sensor nodes;
- Performance evaluation and validation of the proposed solution.

This work of research and engineering is expected to produce not only the desired architecture and application, but also to provide the opportunity to disseminate the knowledge and software through, at least, a conference paper.

1.3. Main contributions

This section is devoted to the scientific contributions of this dissertation to the state-of-the-art in wireless sensor networks and mobile computing.

The first contribution is the proposal and construction of a ubiquitous model that allows the remote monitoring of wireless sensor networks in mobility environments, which is described in Chapter 2. The proposed model allows mobile users to access sensed data from the database in a platform independent way on ubiquitous environments. This model was presented at the *Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2012)*, at Palermo, Italy, July 04-06, 2012.

The second contribution is the proposal, construction, performance evaluation, and validation of an architecture for wireless sensor networks monitoring with push notifications and end-to-end connectivity. This proposal allows users to receive alerts when significant changes occur in the network and also to monitor the wireless sensor network in real-time. This contribution is presented on Chapter 3 and was submitted to an international journal.

1.4. Dissertation Organization

This dissertation is organized in four chapters and the chapters are organized as follows. This chapter, the first, presents the context of the dissertation, focusing the topics under studies, the objectives, the main contributions and the dissertation organization.

Chapter 2 presents the proposal of a model for ubiquitous monitoring of wireless sensor networks based on smartphones, Web services and a relational database. The proposed model allows mobile users to request latest sensed data from the WSN and present it on the mobile device.

Chapter 3 presents a detailed description of a ubiquitous architecture for WSNs monitoring and introduces a real-time monitoring functionality and a push notifications system allowing mobile users to receive alerts when

significant changes occur in the WSN. Moreover, tests and validation of the proposed solution are presented.

Chapter 4 addresses the presentation of the main conclusions of this dissertation and points some directions for future work.

References

- [1] I.F. Akyildiz, Su Weilian, Y. Sankarasubramaniam and E. Cayirci, *A survey on sensor networks*, IEEE Communications Magazine **40**, 2002, pp. 102-114.
- [2] L. Mottola and G. P. Picco, *Programming wireless sensor networks: Fundamental concepts and state of the art* **43**(3), ACM Computing Surveys, 2011.
- [3] J. M. Corchado, J. Bajo, D.I. Tapia and A. Abraham, *Using Heterogeneous Wireless Sensor Networks in a Telemonitoring System for Healthcare*, IEEE Transactions on Information Technology in Biomedicine **14**(2) 2010, pp.234-240.
- [4] L. M. L. Oliveira and J. J. P. C. Rodrigues, *Wireless Sensor Networks: a Survey on Environmental Monitoring*, Journal of Communications **6**(2), 2011, pp. 143-151.
- [5] P. Wang, Z. Sun, M. C. Vuran, M. A. Al-Rodhaan, A. M. Al-Dhelaan and I. F. Akyildiz, *On network connectivity of wireless sensor networks for sandstorm monitoring*, Computer Networks **55**(5), Elsevier, 2011, pp. 1150-1157.
- [6] C. Alcaraz, P. Najera, J. Lopez and R. Roman, *Wireless Sensor Networks and the Internet of Things: Do We Need a Complete Integration?*, 1er International Workshop on the Security of The Internet of Things, 2010.
- [7] L. Atzori, A. Iera and G. Morabito, *The Internet of Things: A survey*, Computer Networks **54**(15), Elsevier, 2010, pp. 2787-2805.
- [8] L. M. L. Oliveira, A. F. de Sousa and J. J. P. C. Rodrigues, *Routing and mobility approaches in IPv6 over LoWPAN mesh networks*, International Journal of Communication Systems **24**(11), Wiley, 2011, pp. 1445–1466.
- [9] A. Ludovici, A. Calveras and J. Casademont, *Forwarding Techniques for IP Fragmented Packets in a Real 6LoWPAN Network*, Sensors **11**(1), 2011, pp. 992-1008.
- [10] D.S. Tudose, A. Voinescu, M. Petrareanu, A. Bucur, D. Loghin, A. Bostan, M. Tapus, *Home automation design using 6LoWPAN wireless sensor networks*, in DCOSS, International Conference on Distributed Computing in Sensor Systems and Workshops, June 27-29, 2011.
- [11] P. Sanyal, S. Das, S.S. Bhunia, S. Roy and N. Mukherjee, *An experience of implementing IPv6 based data retrieval system for Wireless Sensor Networks*, on RACSS, International Conference on Recent Advances in Computing and Software Systems, April 25-27, 2012.

- [12] L. Herrera, B. Mink, S. Sukittanon, *Integrated personal mobile devices to wireless weather sensing network*, Proceedings of the IEEE SoutheastCon, March 18-21, 2010.
- [13] M. Cardei, A. Marcus, I. Cardei, T. Tavtilov, *Web-based heterogeneous WSN integration using pervasive communication*, in IPCCC, IEEE 30th International Performance Computing and Communications Conference, November 17-19, 2011.
- [14] Z. Li, Y. Liu, M. Li, J. Wang and Z. Cao, *Exploiting Ubiquitous Data Collection for Mobile Users in Wireless Sensor Networks*, IEEE Transactions on Parallel and Distributed Systems, Issue 99, 2012.
- [15] R. Kemp, N. Palmer, T. Kielmann and H. Bal, *Energy Efficient Information Monitoring Applications on Smartphones through Communication Offloading*, Mobile Computing, Applications, and Services, **95**(2), Springer, 2012, pp. 60-79.

2. A Ubiquitous Model for Wireless Sensor Networks Monitoring

This chapter consists of the following article

A Ubiquitous Model for Wireless Sensor Networks Monitoring

André G. F. Elias, Joel. J. P. C. Rodrigues, Luís M. L. Oliveira, Bruno B. Zarpelão

Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2012), Palermo, Italy, July 04-06, 2012.

A Ubiquitous Model for Wireless Sensor Networks Monitoring

André G. F. Elias¹, Joel J. P. C. Rodrigues¹, Luís M. L. Oliveira¹, Bruno B. Zarpelão²,

¹ Instituto de Telecomunicações, University of Beira Interior, Portugal

² School of Electrical and Computer Engineering, University of Campinas, Brazil
andre.elias@it.ubi.pt; joeljr@ieee.org; {loliveira; bruno.zarpelao}@it.ubi.pt;

Abstract— Wireless sensor networks (WSNs) belong to emerging technologies where network devices can interact with the surrounding environment by sensing physical parameters. Recently, with the dissemination of mobile devices to Internet connectivity, users can interact with sensor networks and collect environmental data, anytime, anywhere using user-friendly mobile applications. Following the Internet of Things vision, the integration of all sorts of Internet-based devices is considered a big challenge. New infrastructures are required in order to interconnect these devices independently of the used technologies. This paper proposes a model for WSNs monitoring based on a REST Web service and XML messages to provide a mobile ubiquitous approach for WSN monitoring. Data collected from a WSN is stored in a database. Then, mobile clients send XML based messages to a HTTP server through a well-defined REST interface, requesting WSN collected data. A WSN laboratory testbed was used to perform the evaluation, demonstration, and validation of the proposed model. Results show that proposed solution is able to collect and present data in a mobile environment, and it is ready for use.

Keywords - Wireless sensor networks; Ubiquitous computing; Network monitoring; Mobile computing; Web services; Internet of things; IoT

I. INTRODUCTION

Wireless sensors are small and autonomous devices capable of measuring all sorts of environmental and physical conditions. Based on this kind of sensors, a recent network approach, known as wireless sensor networks (WSNs), has become an important field of research. It is now widely used in several areas since it provides a wide range of environment monitoring and military surveillance applications among others [1, 2]. Wireless sensor networks infrastructures are based on several spatially distributed sensor nodes, with the ability to communicate wirelessly. They allow remote monitoring of physical phenomena, such as temperature, humidity, pollution, or natural disasters [3, 4]. One of the main challenges in this field is the connection between these sensor nodes and the Internet. Usually, sensors connect to each other on top of proprietary protocols, because the Internet protocol (IP) protocol is heavy for these tiny devices. The communications between WSNs and the Internet became possible due to the standardization of IPv6 over Low-power Personal Area Networks (6LoWPANs). This technology adds

an adaptation layer below IP and enables the transmission of IPv6 datagrams over IEEE 802.15.4 wireless links [5].

The use of WSNs in the context of ubiquitous computing has emerged in recent years. With an enormous growth of mobile devices and operating systems, these mobile devices, specifically smart phones, were placed as key elements in future ubiquitous wireless networks.

An important new trend in Internet development is heterogeneity. Sensors, smart phones, and tablets are based on different software and hardware platforms and technologies, composing heterogeneous network scenarios. Recently, a new paradigm has emerged where all sorts of devices are connected to the Internet. It is the so-called Internet of Things vision [6] where all the IP-enabled devices connect the Internet independently of the used protocols and communication layers. Therefore, it is necessary to develop models that enable these devices to exchange information independent of their implementation.

This paper proposes a ubiquitous model for wireless sensor networks monitoring through a REST Web service based on open standards such as Hypertext Transfer Protocol (HTTP) and Extensible Markup Language (XML). A database stores wireless sensor networks, which is delivered to mobile clients in XML messages by a HTTP server. The REST Web service is based on a modular architecture and can be upgraded to support more functionalities and resources, which makes the system scalable. The objective is to enable mobile clients and the server to exchange information in a platform independent way.

To evaluate and demonstrate the proposed WSN ubiquitous monitoring model, a laboratory testbed was deployed. This network collects data about air temperature, humidity, and luminosity. It was shown that proposed model is able to efficiently monitor environmental conditions through mobile devices. With this proposal, mobile devices are capable of obtaining sensing information from the server database in real time and also access historical data.

The remainder of this paper is structured as follows. Section II discusses some related work about the topic, while Section III presents the proposal of the overall model architecture. Section IV addresses the construction of the proposed model and the mobile application for Android operating system. Finally, Section V concludes the paper and points further research directions.

II. RELATED WORK

Wireless sensor networks (WSNs) and environmental monitoring require solutions that present sensor data to the user with efficiency and simplicity. With the dissemination of mobile devices, is now possible to use a mobile device as the center of a WSN platform and provide the user with a clean UI in order to see all the information available in the sensor network. This section presents some available projects regarding ubiquitous solutions for monitoring wireless sensor networks.

Li *et al.* [7] proposed an approach for mobile users to collect network-wide data using mobile handheld devices. These devices communicate directly with nearby sensor nodes in the network through IEEE 802.15.4. Unlike static approaches where sensors send data to the sink node, in the presence of user mobility the data collection tree needs to be updated due to the mobility of the user. This functionality reduces the communications delay and provides real-time data collection.

WSN Monitor is a modular solution, presented by Vajsar and Rucka [8], which is able to monitor and manage wireless sensor networks. WSN Monitor is based on database storage, a server software and client applications. The solution is scalable due to its modular architecture build up from modules such as, graphic and sensor modules. The server application accesses data in the database and processes requests from the client. The client application is based on Adobe Air technology, which is supported in some mobile devices but with limitations when compared with a native mobile application. The responsiveness and performance is consistently better in native applications and the user interface has the same look and feel across the system.

A software solution called Sensor Explorer is proposed by [9]. This software is based in a modular design and allows the administration and monitoring of wireless sensor networks. It is capable of presenting a 3D view of the WSN topology. Client applications can run on a large range of platforms including desktops and mobile devices. The data packets collected by sensor nodes are transmitted to the server which provides database storage as well as interfacing to external clients for visualization of collected data and WSN topology.

Parbat *et al.* [10] presented a survey on data visualization tools used in WSN monitoring. In this study, nineteen data visualization tools are presented and analyzed in terms of their user interface, portability and simplicity. Some of the presented tools are reviewed with detail focusing the system architecture, the communication protocols and the monitoring application itself while the most part are presented in a summarized way. Most of these tools are proprietary software and only offer support for specific hardware manufacturers and none of them was built specifically for a mobile environment.

Kim *et al.* [11] proposed an architecture model for WSN's real-time monitoring. The proposed architecture collects data from a TinyOS based WSN and then store the data on a MySQL database that is accessed by the monitoring application. The model uses artificial intelligence algorithms to filter all the collected data before presenting it to the user. A Windows application that communicates with the server over IPv4 was used to visualize the sensed data.

There are several approaches and implementations with the same purpose to control and to monitor a wireless sensor network. This paper proposes a robust and reliable model to provide the user with a mobile platform capable of presenting data in a mobility environment. Besides that, the proposal is designed to be independent of client and server implementations since the core of the model is based on platform independent XML messages and REST interfaces.

III. MODEL ARCHITECTURE

The proposed three-tier architecture is based on the following main components: a relational database, a Web service, and a mobile client application. The entire system architecture is illustrated in Fig. 1. The database was designed to store all the information about a WSN and also user credentials such as username and password. The portability and scalability of the database were key requirements in order to allow its implementation across multiple platforms and database architectures. The structure of the database is a result of several iterations, from the analysis of different WSNs to the study of existing solutions. Based on this analysis, the database stores all the relevant data needed to remotely access and monitor a WSN, such as:

- Information of each node in the network, MAC address, IP address, GPS coordinates, etc;
- Name, value, unit, and timestamp for each mote parameter;
- Mote manufacturer and country of origin;
- Information about the localization and the environment of the WSN deployment;
- Information about user access, username, password, and email.

The communication between the database and the mobile client application is assured through a Web service. As the current mobile devices do not support IPv6 yet, the mobile application and the Web service communicate over IPv4. An Internet connection on the mobile device is required in order to communicate with the server. This interconnection between Internet-based devices is one of the biggest challenges of the Internet as defended by the Internet of Things vision.

Building a Web service on top of industry open standards was a main requirement. This will allow the remote access from multi-platform client applications to the database and also the use of different database systems and architectures. After considering several approaches in Web services development, the Representational State Transfer (REST) architecture was chosen [12]. The REST architecture is based on client-server communication, where clients send requests to the server and the server returns the current state of an available resource. A resource can be defined as an entity or information that is exposed by the service. In a RESTful architecture, the client application uses HTTP protocol methods such as GET and POST to request resources from the server.

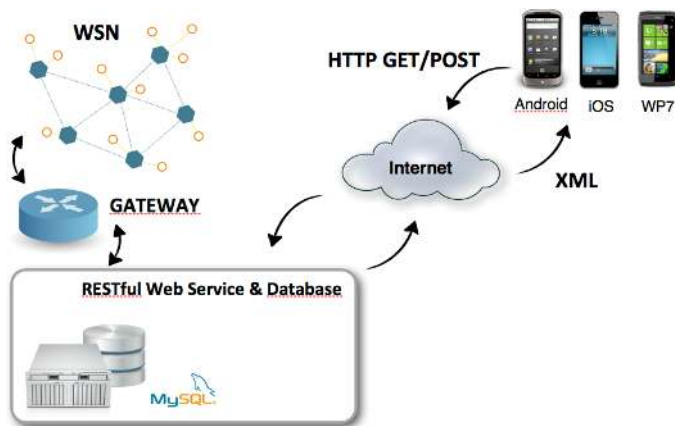


Figure 1. Illustration of the system architecture diagram.

The XML format was chosen to exchange data between the Web service and the client application because it is widely supported in both client and server architectures. Using these technologies, the mobile client application sends an HTTP GET/POST request to the Web service that runs a query in the database and returns the result to the client through an XML file. The XML file structure was defined to be scalable and uniform throughout the system and the mobile applications. The `<result>` tag was chosen to define the result set returned by the server and the `<row>` tag is used to define each individual row of the result set. All other tags derived from the name of each attribute in the database. Thus, an example of an XML file with information about the last temperature reading on a given sensor node would be the following:

```
<?xml version="1.0"?>
<result>
  <row>
    <value>27.14</value>
  </row>
</result>
```

The proposal join generic messages in XML and well defined REST interfaces to build a communication protocol that enables client and server to exchange messages in a platform independent way.

The integration of these technologies allows users to perform various operations on the mobile application. Two major operations are server database real-time access and historical data queries. The mobile application sends requests to the Web service repeatedly with a time interval defined by the user. The same applies to the database that is updated with the last readings from the sensors with a previously defined frequency. In order to access historical data, the mobile application sends a request to the web service with a chosen time period. The Web service returns an XML file with the sensor readings within that time period and the mobile application presents this data graphically.

IV. CONSTRUCTION OF THE PROPOSED MODEL

The model has been created on top of open standards that make the system more generic, portable, and scalable. A real environment has been created with all the components needed for a full WSN monitoring solution.

A. Data Storage

The MySQL Database Management System (DBMS) was chosen for data storage. MySQL is an open source technology and provides performance and scalability to the system. The support for the majority of operating systems is also an advantage of this technology. The performance was an important issue in the construction of the database due to the large amount of data collected. The number of records increases exponentially depending on the updates frequency.

The database was designed in order to reduce redundant and null values and also to optimize the performance. The structure of the database is based in nine related tables that can be divided into three groups accordingly to the data type of each table. The tables *user* and *groups* can be grouped as they both relate to the users and their permissions. Another group store information about sensor nodes and their location. This group of tables store specific information of each network node such as the IP address and the GPS coordinates as well as the type of sensor modules available on each node. Finally, another group of tables store all the collected data by the sensor nodes for each type of sensor.

In order to access collected sensor data, parsing is needed in the bad-end in order to convert this information into a readable format. Collected data are sent to a gateway computer and parsed into a comma-separated values (CSV) file through an algorithm that analyze and separate the raw byte stream sent by each mote in the network [13]. Then, the CSV file is used to store the data in the database.

B. Web Service

To enable the interaction between the database and the mobile devices, a Web service was built in order to expose the database through a modular and generic Application Programming Interface (API) that can be accessed through several mobile platforms. After analyzing various approaches in Web service development, the REST architecture was chosen. This architecture follows the client-server model and is based on the HTTP protocol that is supported by almost all mobile devices with an Internet access.

Thus, the Web service was developed using the Jersey open-source framework [14] that is the reference implementation in the construction of RESTful web services. The web service was built using a modular approach to facilitate its subsequent maintenance and scalability. It considers a three-tier approach with the following modules: the database module, the resource module, and the parsing module. The database module manages all the connections to the database while the resource module represents the resources that are identified with a unique Uniform Resource Identifier (URI) and can be accessed through an HTTP request. The following URI, "http://[server-ip]/rest/sensors" is an example of an HTTP request that is sent from the mobile device to the web service and requests information about the

available sensors on a network. The parsing module converts a result set from the server database into a specific message format, in this case, the XML based format defined in our model. With this modular structure, the Web service can be easily reused and deployed in another environment or platform without compromising its scalability.

V. PERFORMANCE EVALUATION AND DEMONSTRATION

In order to evaluate and demonstrate the proposed model, a testbed was created. A 6LoWPAN WSN was created in order to collect temperature, humidity, luminosity, and battery voltage readings. Furthermore, a mobile Android application was built to consume the REST services, presenting the collected data to the user in an easy and meaningful way.

A. Wireless Sensor Network Testbed

In the design of the 6LoWPAN wireless sensor network, eight motes running the TinyOS operating were used. This network may be seen in Figure 2. These motes communicate through IEEE 802.15.4 and the 6LoWPAN protocol stack is provided by the TinyOS Blip 1.0 implementation. The motes are capable of sensing air temperature and humidity, luminosity and battery voltage readings. A 6LoWPAN gateway is used to provide IPv6 end-to-end connectivity between the sensor network and the Internet. The 6LoWPAN gateway runs on Ubuntu 10.0.4 and it has multiple communication interfaces technologies, including IEEE 802.15.4, Ethernet and IEEE 802.11a/b/g. To implement the IEEE 802.15.4 interface a TelosB mote connected to an USB port was used. An Intel desktop board D945GCLF with a 1.6 Ghz Intel Atom processor has been used to be the motherboard of the gateway. The application IP-driver compliant with RFC 4944, provided by TinyOS 2.1, act as the 6LoWPAN adaptation layer in the gateway. The 6LoWPAN gateway is also responsible for announcing the IPv6 prefix and the default gateway address to all sensor nodes.

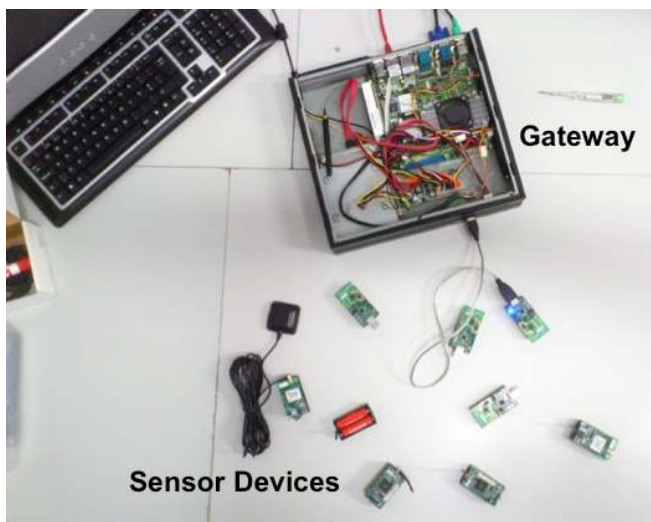


Figure 2. Photo of the used WSN laboratory testbed with sensors and the network gateway.

B. Results

For evaluation, demonstration and validation purposes, a mobile Android application was built. The Android OS is an open-source mobile operating system supported by Google and based on the Linux kernel. The Android System Development Kit (SDK) is built in Java language and provides the necessary libraries and tools to write native Android applications and deploy it to the device.

The mobile Android application was developed to communicate to REST interfaces that deliver WSN data stored in the MySQL database. Considering the developed architecture, the requirements analysis was focused on the features that could be provided by the model. It was defined that mobile application should perform two basic operations, real-time access to the server database and historical data query. Secondary requirements were also defined, such as user authentication, graphical data presentation, visualization of several measures simultaneously, and the possibility to choose the time interval of the updates.

The user interface of the Android application was designed to be user-friendly and to present data to the user in a clean and organized way. The application is based on two types of navigation, tabbed navigation and hierarchical navigation. The user can select from three different tabs, the sensors tab, the historic tab, and the settings tab. Within each tab the navigation is hierarchical.

Figure 3 presents the initial login screen and the main data visualization screen. Following Fig. 3, label “1” points out the login screen where the user can enter his/her username and password to log in the system. If the user authentication is successful, a screen with the three tabs is presented where the sensors tab is selected by default as shown by label “2”. Initially, the sensors tab shows a list of the available WSNs that user can access depending on the permissions of his/her user group. When a user selects one of the available WSNs, a new list is shown with the names of the sensor nodes that belong to this network. On this screen, if the user chooses one of the sensor nodes, the data visualization screen is presented and shows the latest available measurements for that sensor in both scalar and graphical modes. Label “3” points out the scalar data presentation that shows four sensor readings of the same mote simultaneously. If a mote has more than four sensor modules the user can use the button pointed out by “4” to choose which are the four sensor values that he wants to see on the screen at the same time. The collected data is also presented in a graphical mode as pointed out by “5”

The history screen and the settings menu are presented in Fig. 4. If a user selects the history tab pointed out by “1”, a new screen is shown and it allows users to choose a time interval that may be days or only a few hours. Labels “2” and “3” identify the interface buttons to select the time interval. After choosing the time interval, if the user taps the View Graph button, identified by label “4”, a full-screen graph is shown with all the measures collected on that time interval. In order to change the default application settings, a settings menu was created. Label “5” points out the settings tab that allows the user to enable or disable data updates in order to reduce power



Figure 3. Login and data visualization screens.

and bandwidth consumption. This setting is enabled by default as shown by “6”. When enabled, the user can choose the frequency of the updates between predefined values from 1 second to 1 minute by selecting the option pointed out by “7”. The mobile Android application retrieved sensor data by sending XML based requests to the REST interfaces deployed on the HTTP server. The application was evaluated and experimented in several devices with different specifications and screen sizes. The version 2.3 of the Android OS was used to develop the application but it also runs on previous versions. The tests proved that the application is reliable and runs smoothly in a large range of devices.

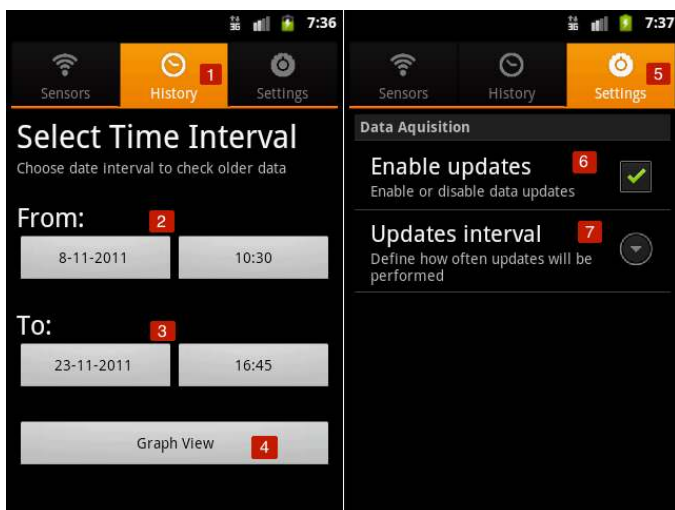


Figure 4. History and Settings screens.

VI. CONCLUSIONS AND FUTURE WORK

This paper proposed a monitoring model to interact with wireless sensor networks. The model allows users to remotely monitor the network, obtaining most recent sensor readings and access to historical data. Based on open standards and constructed with a modular architecture, the solution is reliable and scalable. Besides, the employment of generic XML messages and REST well-defined interfaces allows

mobile clients and servers to exchange messages in a platform independent way. This interconnection between different platforms and architectures through the Internet follows the Internet of Things vision. In order to evaluate and demonstrate the proposed model, a wireless sensor network testbed was deployed. The mobile client application was tested through different mobile devices with success and is ready for use.

As future work, the proposed model may be extended outside the laboratory for real environment monitoring. In this case, issues such as power management and security must be considered. Furthermore, a push notification system may be an improvement to the model in order to alert the user if a sensor reading overcomes a given threshold.

ACKNOWLEDGMENTS

This work has been partially supported by the *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Portugal, and by National Funding from the FCT – *Fundação para a Ciência e Tecnologia* through the Pest-OE/EEI/LA0008/2011 project.

REFERENCES

- [1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor network: a survey”, *IEEE Communications Magazine*, Vol. 40, pp. 102–114, 2002.
- [2] L.M.L. Oliveira and J.J.P.C. Rodrigues, “Wireless Sensor Networks: a Survey on Environmental Monitoring”, *Journal of Communications (JCM)*, Academy Publisher, Vol 6, No 2, pp. 143-151, April 2011.
- [3] L. Yu, Q. Zhang, X. Meng, and Z. Yan, “Design of the granary temperature and humidity measure and control system based on Zigbee wireless sensor network”, *International Conference on Electrical and Control Engineering (ICECE 2011)*, Sept. 16-18, 2011.
- [4] N.A.A. Aziz and K.A. Aziz, “Managing disaster with wireless sensor networks”, *International Conference on Advanced Communication Technology (ICACT 2011)*, Feb. 13-16, 2011.
- [5] L. M. L. Oliveira, A. F. de Sousa, and J. J. P. C. Rodrigues, “Routing and mobility approaches in IPv6 over LoWPAN mesh networks”, *International Journal of Communication Systems*, Wiley, ISSN 1074-5351, Vol 24, Issue 11, pp. 1445–1466, November 2011.
- [6] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey”, *Computer Networks*, Vol 54, No 15, pp. 2787-2805, October 2010.
- [7] Z. Li, M. Li, J. Wang, and Z. Cao, “Ubiquitous data collection for mobile users in wireless sensor networks”, *IEEE INFOCOM 2011*, April 10-15, 2011.
- [8] P. Vajsar and L. Rucka, “Monitoring and management system for wireless sensor networks”, *34th International Conference on Telecommunications and Signal Processing (TSP 2011)*, Aug. 18-20, 2011.
- [9] R. ElHakim, A. Orfy, and M. ElHelw, “Monitoring and visualization of large WSN deployments”, *Sensors, IEEE*, pp.1377-1381, Nov. 1-4, 2010.
- [10] B. Parbat, A.K. Dwivedi, and O.P. Vyas, “Data Visualization Tools for WSNs: A Glimpse”, *International Journal of Computer Applications (IJCA)*, Vol 2, No 1, May 14-20, 2010.
- [11] D. Kim, B. An, and N. Kim, “Architecture Model of Real-time Monitoring Service Based on Wireless Sensor Networks”, *10th International Conference on Advanced Communication Technology (ICACT 2008)*, Feb. 17-20, 2008.
- [12] F. Belqasmi, R. Glitho, and C. Fu, “RESTful web services for service provisioning in next-generation networks: a survey”, *Communications Magazine, IEEE*, Vol 49, No 12, pp. 66-73, December 2011.
- [13] B. da Silva Campos, J.J.P.C. Rodrigues, L.D.P. Mendes, E.F. Nakamura and C.M.S. Figueiredo, “Design and Construction of Wireless Sensor Network Gateway with IPv4/IPv6 Support”, *International Conference on Communications (IEEE ICC 2011)*, pp. 1-5, June 5-9, 2011.
- [14] “Jersey Project”, Oct. 2011. [Online]. Available: <http://jersey.java.net>

3. Ubiquitous Monitoring Solution for Wireless Sensor Monitoring with Push Notifications and End-to-End Connectivity

This chapter consists of the following article

André G. F. Elias, Joel. J. P. C. Rodrigues, Luís M. L. Oliveira, Bruno B. Zarpelão

Scientific paper submitted to an international journal.

Ubiquitous Monitoring Solution for Wireless Sensor Networks with Push Notifications and End-to-End Connectivity

André. G. F. Elias¹, Joel J.P.C. Rodrigues¹, Luis M. L. Oliveira¹, Bruno B. Zarpelão²

¹Instituto de Telecomunicações, University of Beira Interior, Portugal

²Pontifícia Universidade Católica do Paraná (PUC-PR), Londrina, Brazil

andre.elias@it.ubi.pt; joeljr@ieee.org; {loliveira; bruno.zarpelao}@it.ubi.pt

Abstract – Wireless Sensor Networks (WSNs) belongs to a new trend in technology in which tiny and resource constrained devices are wirelessly interconnected and are able to interact with the surrounding environment by collecting data such as temperature and humidity. Recently, due to the huge growth in the use of mobile devices with Internet connection, smartphones are becoming the center of future ubiquitous wireless networks. Interconnecting WSNs with smartphones and the Internet is a big challenge and new architectures are required due to the heterogeneity of these devices. Taking into account that people are using smartphones with Internet connection, there is a good opportunity to propose a new architecture for wireless sensors monitoring using push notifications and smartphones. Then, this paper proposes a ubiquitous approach for WSN monitoring based on a REST Web Service, a relational database, and an Android mobile application. Real-time data sensed by a WSN is sent directly to the smartphone or stored in the database and requested by the mobile application using a well-defined RESTful interface. A push notification system was constructed in order to alert mobile users when a sensor parameter

overcomes a given threshold. The proposed architecture and mobile application were evaluated and validated using a laboratory WSN testbed and are ready for use.

Keywords – Wireless Sensor Networks; Internet of Things; Ubiquitous computing; Network monitoring; Mobile computing; Restful Web Services; Push notifications; Android applications

1. Introduction

Wireless sensors are tiny devices that are able to measure several environmental and vital variables. Recently, these devices have been used in areas such as environmental monitoring, home automation, and war scenarios. In this context, a new emerging technology called wireless sensor networks (WSNs) has become a trend in technological research [11, 19]. This technology combines hundreds or even thousands of tiny and resource constrained sensor devices that communicate wirelessly in order to accomplish a common task. These devices are spatially distributed in the environment in order to collect data about surrounding environmental variables [17, 27]. Each device has several sensor modules capable of measuring parameters such as temperature, humidity and luminosity.

The main challenges regarding wireless sensor networks are power consumption of sensor devices and their connection to the Internet [8]. Power consumption is highly affected by the communication between nodes. One solution to this problem is reducing the communication between nodes using better routing algorithms and shutting down the nodes when they are not required [18, 23].

Connecting these limited devices to the Internet is a big challenge because the use of the TCP/IP stack, as initially conceived, is too heavy in the context of WSNs. However, the use of IPv6 over Low-power Personal Area Networks (6LoWPANs) allows the transmission of IPv6 packets over IEEE 802.15.4 wireless links and enables WSNs to communicate with the Internet more efficiently [31]. Basically, 6LoWPAN adds an adaptation layer below network layer that fragments the packets and compresses the IPv6 transport layer headers. These adjustments allow the IPv6 to be used in low-power networks such as WSNs. The connectivity between

6LoWPAN-enabled WSNs and IPv6 networks is not straightforward, because devices with IPv6 support are not able to handle the 6LoWPAN compression and fragmentation [2]. One solution to this issue is using an intermediary element such as a gateway that allows data exchange between WSNs and IPv6 hosts through 6LoWPAN [7, 25].

The introduction of 6LoWPAN made IPv6 suitable for resource-constrained devices enabling their connectivity to the Internet. Smartphones are becoming current personal computers, but it is a big challenge to establish end-to-end connectivity (between an end-user device and a sensor node) with all sorts of smart objects as defended by the Internet of Things vision [15]. To solve this issue, an end-to-end connectivity solution is proposed in order to interconnect smartphones and sensor devices allowing real-time mobile monitoring.

The integration between these mobile devices, specifically smartphones, and future wireless networks is crucial for the Internet growth. In the context of WSNs, the use of smartphones to control and monitor these kinds of networks represents a new trend in ubiquitous computing research [32]. The growing diversity of mobile operating systems and hardware platforms is developing new and heterogeneous network scenarios. Therefore, the construction of new models and architectures to enable the interaction between these devices in a platform independent way is essential [12].

Due to the heterogeneity of recent mobile devices and platforms, the construction of a Web service to interconnect these devices in a platform independent way require open technologies and protocols. The interaction between smart phones and WSN devices is possible by using Web technologies such as Web services and IP-enabled

devices. The Representational State Transfer (REST) architecture is commonly used in the construction of Web services since it is based on HTTP that is supported by all smartphones [29]. The REST architecture is based on client-server communication, where clients request available resources from the Web service. A REST resource is defined as a set of data that is available through a well-defined RESTful interface [10]. Mobile clients request these resources using well-known HTTP methods such as GET and POST. To exchange information between the REST Web service and the mobile device, XML, and JSON media types are commonly used. Furthermore, since a WSN involves large amounts of data, the exchange of information needs to be optimized.

Wireless sensor networks are strictly related to monitoring solutions and the information collected by the sensors is more important than the sensor itself. In order to increase the efficiency of wireless sensor network monitoring, the user should be alerted when there are significant changes in sensed data. In a mobile environment, it is becoming natural for us to receive alerts or notifications from several services such as email and newsletters. The integration of push-notifications in ubiquitous wireless sensor networks makes sense since it is crucial for one to be alerted when a value collected by a sensor exceeds a threshold. Recent mobile operating systems are also capable of receiving and presenting this type of notifications seamlessly. Pushing notifications to the mobile device represents significant energy saving when compared with always-on solutions based on polling requests [28].

The main contributions of this paper are the following: (1) the introduction of a system architecture specifically designed to collect, store, and present real-time wireless sensor networks data and historical measures in mobile environments; (2) a

push notification system that allows mobile users to receive an alert over the Internet when a sensor value overcomes a given threshold; (3) an Android mobile application that presents the latest data collected by the WSN and historical measures in scalar and graphical ways; (4) and the testbed architecture definition to evaluate the proposed approach.

This paper is organized as follows. Section 2 reviews available related literature about the topic. Section 3 introduces and describes the overall system architecture and the interaction between its modules. Section 4 addresses the implementation of the proposed architecture while Section 5 presents the design and construction of the mobile application. Section 6 evaluates the proposed solution considering its architecture and mobile application with measurements of example scenarios. Finally, conclusions are presented in Section 7.

2. Related Work

WSN monitoring applications demand new functionalities and design patterns to address recent challenges in efficiency, interoperability and user interaction. The increasing heterogeneity of mobile devices, operating systems, and communication interfaces requires modern architectures and mobile applications that access the information in a platform independent way. This section presents some available solutions regarding WSN monitoring. Some projects address the used communication protocols and their architectures while others focus on the application layer.

Munawar *et al.* [3] proposed an Open Sensor Platform to interconnect mobile devices and sensors. The solution uses available commercial hardware and software tools such as a proprietary data acquisition device. This device receives sensed data and sends the information to a host PC using a proprietary PC application. The information is then requested by the mobile device through a Symbian OS application. Since the presented solution does not use the Internet to retrieve data from the sensors, the mobility of the user is highly reduced. Also, using proprietary hardware and software is a limitation due to interoperability and costs. The use of Symbian OS is also a limitation, since it is becoming obsolete when comparing to modern mobile operating systems such as Android and iOS.

Herrera *et al.* [16] present an approach to wireless sensor networks monitoring using Zigbee and the iPhone platform. The focus of this solution is to collect environmental and weather data from remote stations and present the information on the mobile device. The remote stations are equipped with a microcontroller that periodically gathers data from the sensors such as wind speed and rainfall. The sensed data is then parsed into a readable format and sent to a proprietary gateway coordinator. This coordinator interconnects the Zigbee personal area network and other network with access to database servers. The data is stored in a MySQL database through PHP scripting and a web page. The mobile application sends requests to the web page in order to retrieve last sensed data. The use of PHP and HTML instead of a structured Web Service is a limitation of this approach in terms of performance and scalability.

Hornsby *et al.* [1] proposed an architecture that is based on the XMPP-protocol and wireless sensor networks that support a push based notification functionality. The

proposed architecture uses Atom feeds [21] to communicate with the wireless sensor network in a Web Service-like way and an UPnP gateway was used to interconnect the XMPP-based WSN and UPnP media-oriented networks. Therefore, messages are exchanged over IP protocol using the XML media type. In order to interact with the WSN, a solution was implemented in Internet tablet devices. This architecture has some drawbacks related to the use of XMPP-protocol that is not yet supported by many mobile operating systems. This protocol is also restricted to XML media type that is a limitation in modern mobile environments where the use of JSON is becoming a standard.

Cardei *et al.* [20] presented a RESTful architecture for healthcare patient monitoring using heterogeneous wireless sensor networks. This innovative approach uses an Android smartphone as a gateway in order to interconnect the WSN and the Internet. The heterogeneous WSN used in this system interconnects environmental, medical, and smartphone internal sensors. The sensed data is processed on the smartphone and sent to the Internet Web Service using the cellular network. This approach represents a limitation when using the mobile device as a gateway due to the limited resources of these devices, such as processing power and battery consumption.

A monitoring platform, called WSN Monitor, was proposed by Vajsar and Rucka [26]. The platform is based on client-server architecture and is focused on monitoring and managing wireless sensor networks. The collected data is stored in a MySQL database and the server processes requests from the client application using the data available in the database. The client application was developed using the proprietary Adobe Flex framework [4] that has some limitations in terms of performance and support when compared with native mobile applications.

Moreira *et al.* work [24] focuses on the design and construction of a mobile monitoring application for wireless sensor networks. The proposed architecture is based in REST interfaces and XML messaging. The mobile application was built on the top of the Android platform but no performance tests were conducted to validate the architecture and the mobile application. The proposed approach also features an alert functionality that alerts the user when a sensor threshold value exceeds a defined limit. The alerts are sent via SMS or e-mail and not as native push-notifications that are standard across the mobile operating system.

Tudose *et al.* [9] proposed a solution for home automation using a 6LoWPAN wireless sensor network and a mobile monitoring application. The architecture is based on a wireless sensor and actuator network with energy harvesting capabilities that minimize node power consumption. In order to interconnect the WSN and the Internet, a gateway was developed. Periodically, sensed values are transmitted wirelessly to the gateway over a UDP over IPv6 connection. The gateway receives and stores the collected data sensed by WSN devices. The Android mobile application sends REST-like requests to the gateway that responds with data in JSON format. The inexistence of a structured Web Service on the proposed architecture has limitations in terms of scalability and performance. Moreover, since no database was constructed in order to store collected data, the access to historical measures is limited.

In the past few years, several solutions were proposed to control and monitor wireless sensor networks. This paper proposes a reliable architecture to collect, store and present data gathered by wireless sensors in a ubiquitous environment. The proposed architecture is designed to be platform independent on the client and server

sides since it is based on REST interfaces and XML/JSON messages. Besides, the native push-notification functionality presented sends a notification to the mobile user when a sensor reading overcomes a given threshold.

3. System Architecture

The proposed four-tier architecture is based on the following main components: a Web service, a gateway, a relational database, and a mobile client application. The entire system architecture is illustrated in Fig. 1. The communication between these four components allows mobile client devices to present data collected by the sensors in a ubiquitous environment. The architecture was designed to enable the mobile application to present real-time data as well as stored historical measures. A push notification system integrated into the Web service and the mobile application is used to alert the user if a sensor reading overcomes a given threshold.

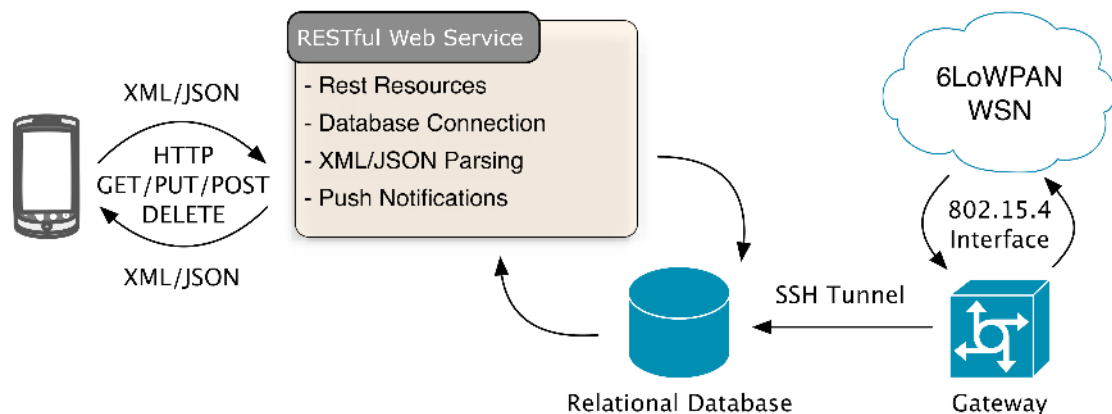


Figure 1. System architecture diagram.

Sensor devices send collected data to the gateway computer in defined time intervals according to the variation of the physical sensed phenomenon. The software in the gateway parses the raw data and stores it in the relational database. Then, the mobile client application sends requests to the Web service that runs a query in the database and returns the result to the mobile application through a specified file format. Besides that, the mobile application can also request real-time data from the gateway. When a sensor reading overcomes a given threshold, the Web Service sends a push notification to the mobile application in order to alert the user that an event occurred. The push notification system is built for the Android operating system and is integrated in Google Cloud to Device Message servers that enable the mobile device to receive push notifications in a mobile environment.

The Web service ensures the communication between the relational database and the mobile application, using HTTP over an IPv4-based network. In order to communicate with the server, an Internet connection is required in the mobile device. This interconnection between Internet-based devices is crucial in the near future and is one of the biggest challenges of the Internet as defended by the Internet of Things paradigm. Designing and building the Web Service on top of industry open standards was a main requirement. Simplicity, scalability and interoperability were also key requirements in the Web service development process. The interoperability ensures the ubiquitous access to the database from multi-platform client applications and also the use of different file formats in the exchange of information. After considering several approaches in Web Services development, the Representational State Transfer (REST) was chosen.

REST is a new architecture for Web services that is being widely adopted by major technology companies. Most of these companies rely on REST for sharing information and to expose web applications and services. REST architecture is based on client-server communication, where clients initiate requests to servers that process these requests and return the appropriate results. These results are defined as resources and represent the information exposed by the service. RESTful architectures are based on HTTP to communicate over the network. HTTP is the protocol of the Web and it has a set of tools that simplify communications such as Uniform Resource Identifiers, request and response headers and Internet media types. These functionalities allow the mobile client application to use the HTTP methods GET, POST, PUT and DELETE to communicate with the Web service and also to exchange information in several file formats such as XML and JSON.

In a mobile environment, 3G Internet connections have several limitations such as bandwidth, speed and cost. Therefore, it is crucial to minimize data traffic in mobile applications and optimize communication protocols. The use of XML and JSON in Web services is the standard and these file structures are widely supported in both client and server architectures. A uniform file structure was defined for both XML and JSON media types in order to be parsed independently by the mobile application. The *result* tag or name was chosen to define the returned result set by the Web service and the *row* tag is used to define each individual row from the result set.

The historical information presented in the mobile application is stored in a relational database. The database was designed specifically to store all the information about a WSN and also user credentials to enable access control and manage user permissions. The scalability and flexibility were key requirements in the design of

the database in order to allow its implementation across multiple server platforms and architectures. The structure of the database is a result of several iterations, from the analysis of different WSN to the study of existing solutions. Based on this analysis, several fields of the database were defined to store all the relevant data needed to remotely access and monitor a WSN. Some of these fields are the following: MAC address, IP address, GPS coordinates; name, value, unit and timestamp for each mote parameter; mote manufacturer and country of origin; information about the localization and the environment of the WSN deployment; and information about user credentials.

In order to get the collected data from the 6LoWPAN WSN and store it in the database, a gateway is needed. The gateway can have more than one IPv6 interface and at least one 6LoWPAN interface to allow the communications between the regular IPv6 node and the WSN. The requests destined to WSN nodes are forwarded to the tap virtual interface and then sent to the 6LoWPAN adaptation layer. The 6LoWPAN adaptation layer is responsible for the packet fragmentation and reassembly, in order to support the IPv6 minimum MTU, and for IP and UDP header compression. The gateway communicates with the 6LoWPAN WSN through an IEEE 802.15.4 sensor node connected to the gateway computer through a USB port. Sensed data received by the gateway is stored in the database through a JDBC connection over an SSH tunnel.

The proposed architecture joins generic messages in XML and JSON, and well-defined REST interfaces to build a communication protocol that enables clients and servers to exchange messages in a platform-independent way. A push notification system in the context of WSN is also proposed and allows mobile users to receive

alerts without requesting them to the server continuously. Besides, the user can also access real-time data instead of getting latest collected values from the database. For real-time monitoring the mobile application sends requests over the Internet to a HTTP service running on the gateway computer that query the WSN and responds directly to the smartphone. The real-time HTTP service acts as an intermediary between the 6LoWPAN wireless sensor network and the IPv4-enabled smartphone.

4. Construction of the Proposed Model

The proposed model architecture was constructed in a real environment with all the needed components for a full WSN monitoring solution. In this section, the development process of the server-side components of the proposed model is described, including a database, a Web service, a push notifications system, and an end-to-end connectivity between mobile devices and sensor nodes.

4.1 Database Design

Based on the requirements analysis, the MySQL Database Management System (DBMS) was chosen for data storage [22]. MySQL provides scalability, flexibility, and is open source. It has support for almost all the operating systems and also provides drivers, plugins, and connectors for the majority of platforms and programming languages.

In the context of WSNs, the performance of the database is an important issue due to the large amount of data collected by the sensors. The number of records and queries increases exponentially depending on the frequency of sensor readings. Tables and

relations were defined accordingly to the requirements analysis based on several WSNs and existing monitoring solutions. Reducing redundant and null values was a main requirement in order to optimize performance and consistency.

The database structure is based on eleven related tables that can be divided into three groups according to the type of stored data. The entity-relationship diagram of the database is illustrated on Fig. 2. The tables *group*, *user*, and *credential* are grouped together because of the relation to user credentials and their permissions. The *user* table stores information about user credentials such as username and password.

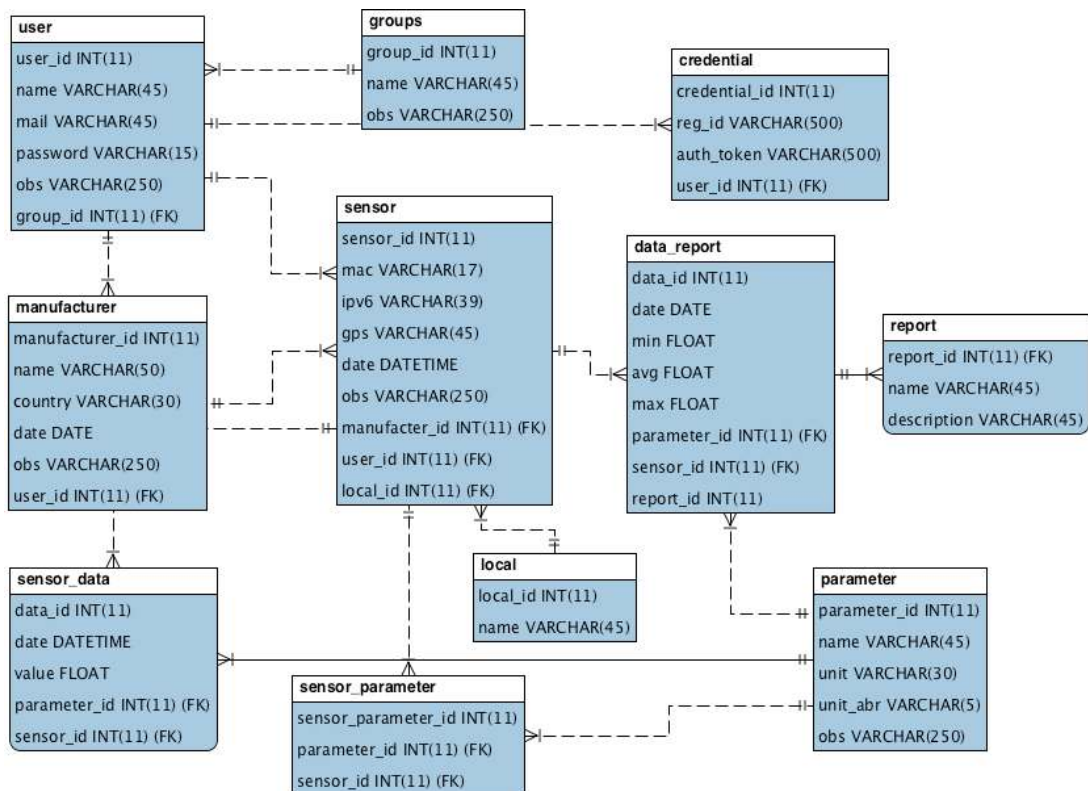


Figure 2. Database Entity-Relationship diagram.

The *group* table stores information about groups of users that have different access permissions to available WSNs while the table *credential* stores the registration ID and the authentication token that enables the mobile application to receive push notifications. The following tables represent another group: *manufacturer*, *local*, *sensor*, *sensor_parameter* and *parameter*. These tables store information about sensor nodes, their location, specifications, and the parameters available in each mote. The *manufacturer* table holds information related to each mote specifications as well as the mote manufacturer.

The *local* table stores information about the geographical location of the WSN. Similarly, *sensor* and *parameter* tables store data related to each individual mote such as the mote's IP address, GPS coordinates and the type of sensor parameters available in each mote. The third group of tables is formed by the tables *sensor_data*, *report* and *data_report* that store the collected data by the sensors for each parameter as well as the maximum, minimum and average values for each sensor parameter grouped by day, month or year. The data stored in the table *data_report* are added through the MySQL event scheduler that calculates the minimum, maximum and average values for each sensor parameter. These events are scheduled to work after each day, month and year.

4.2 REST Web Service

To enable the exchange of information between the database and the mobile devices, a RESTful Web service was built. The Web service has a modular architecture and generic implementation in order to be scalable and accessed through several mobile

platforms. A REST Web service can be defined as a set of resources, available through HTTP interfaces accessed using well-defined HTTP methods such as GET and POST.

Therefore, any client device with an Internet connection and HTTP support can send requests to the Web service in a platform independent way. In the construction of the Web service, the Jersey open-source framework [14] was used. Jersey is the reference implementation for the JAX-RS specification [13] provided by the Java EE 6. It implements the annotations presented on the specification providing a Java API for RESTful Web services development. The modular architecture of the Web service provides scalability to the entire model and it can be further expanded by adding new modules and functionalities without changing the existing ones.

The Web service considers a four-tier approach with the following modules: the database module, the resource module, the parsing module and the notifications module. The Web service architecture may be seen in Fig. 3.

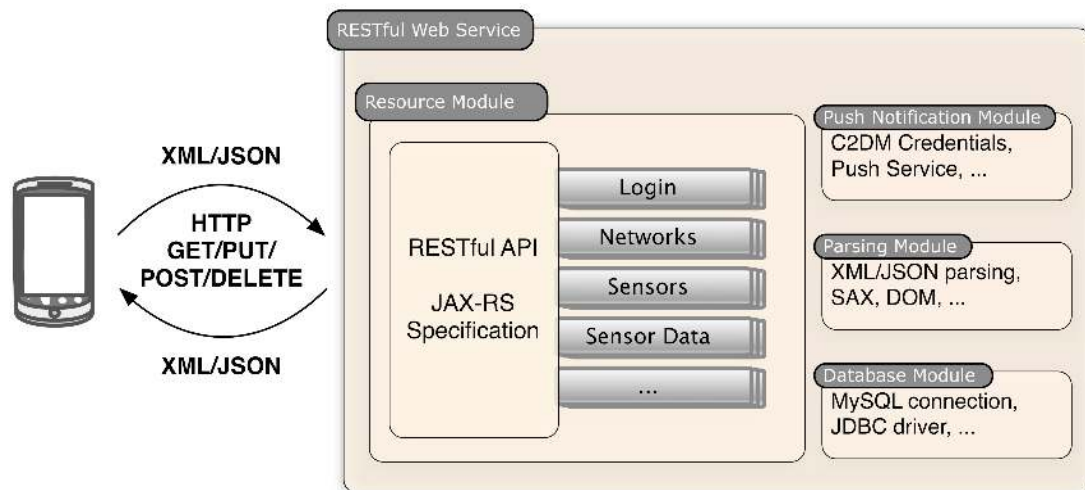


Figure 3. RESTful Web service architecture.

The database module manages all the connections to the database using the Java Database Connectivity (JDBC) API. The RESTful resources are available in the resource module and a unique Uniform Resource Identifier (URI) identifies each resource. A representation of the current state of each resource and a data format known as media type are used to exchange information in the RESTful environment. For example, the following uniform resource identifier (URI), “http://[server-ip]/rest/sensors” is used to request a list of the available sensors in a chosen WSN. When the Web service receives a request of an available resource, an SQL query is sent to the database and then, the parsing module of the Web service converts the result set into the requested media type and returns the resource representation to the client. The client may request the resource representation in XML or JSON media types.

4.3 Push Notifications

One of the main features of the proposed WSN monitoring architecture is the push notification system. This technology is capable of sending messages to the mobile device without constantly poll the server for updates. Polling the server for updates has several well-known issues including the adjustment of the requests frequency, energy consumption and high data traffic. These limitations have more impact when working with mobile devices due to limited battery life and network coverage.

The push notification system is focused on the Android operating system because it is an open platform and integrates well the RESTful Web service and the mobile application. A new technology called Cloud to Device Messaging (C2DM),

developed by Google, was used in the construction of the push notification system [5]. The C2DM technology is part of the Android platform and provides libraries and APIs for developing push-enabled applications.

The integration of push notifications in the WSN monitoring model enables the mobile Android application to receive messages and alert the user when a sensor reading overcomes a predefined threshold. The push messaging technology follows a publish/subscribe model where mobile users register at the server in order to receive push messages on the mobile device. Fig. 4 presents a sequence diagram of the push notification system. To use C2DM, the mobile application must register at the Google authentication servers using a Google account.

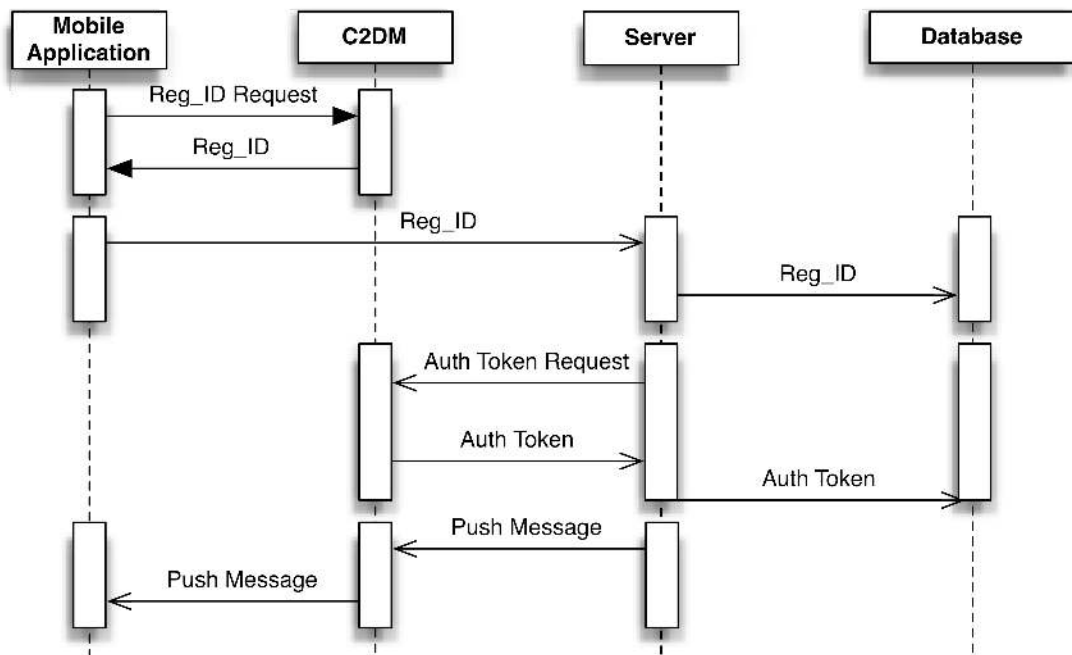


Figure 4. Sequence diagram of the push notification system.

Then, a unique registration ID is generated by the authentication server and sent to the mobile device. This registration ID is forwarded to the Web service and stored in the database. The Web service also requests an authentication token that is used to send notifications to registered mobile devices. Both device registration and server authentication must be completed before sending push messages. Since the push notification system is integrated in the RESTful Web service, it polls the database continuously and when a sensor reading overcomes a predefined threshold, a push message is sent to Google C2DM servers and then to mobile clients. By doing this, the computational costs of polling are on the server side instead of stressing the mobile devices, resulting in significant energy and bandwidth savings. The push notification module was developed in Java language in order to be platform independent as well as the Web server. Therefore, several classes were added to the Web service to support three main functionalities: receiving the registration ID from mobile devices, server authentication and sending push messages.

4.4 End-to-End Connectivity

In order to access real-time data from the wireless sensor network, a multiplatform software application was developed and deployed in the gateway computer enabling the mobile users to request sensor measures. When the mobile client application requests data from the 6LoWPAN WSN, the software application deployed in the gateway computer handles the HTTP request and retrieves data directly from the WSN using the UDP transport protocol. The requested data is transmitted through the IEEE 802.15.4 interface, connected to the gateway via USB. Then, the gateway

application converts the collected data to XML format and forwards it to the smartphone over HTTP using an IEEE 802.11g wireless network with Internet connection. On the client side, the Android application parses the received data and presents it to the user.

5. Android Application

The Android OS is an open mobile operating system, developed and supported by Google. It was built from scratch, specifically for mobile devices and is based on Linux kernel. The Android platform is supported by a wide range of mobile devices, from smartphones to tablets. The Android System Development Kit (SDK) provides libraries and APIs that enable developers to create Android applications and take advantage of hardware capabilities available on the devices using Java programming language. Through the APIs, developers can use functionalities such as text messaging or accelerometers in order to build richer and immersive applications. Since Android is an open platform, it integrates well with emerging technologies and Web services.

5.1 Android User Interface

The user interface was designed following the Android User Interface Guidelines [6] in order to be consistent with the operating system interface and other Android applications. A user-friendly and organized interface was a main requirement in the design process. The application follows two Android navigation guidelines: tabbed navigation and hierarchical navigation. Three fixed tabs at the top of the screen

represent the tabbed navigation and within each tab the relationship between different screens is hierarchical.

The initial screen of the application is the login screen, as seen in Fig.5. In this screen, the user must authenticate with username and password in order to access the application's main functionalities. If the user authentication is successful, a new screen is presented with the tab bar at the top where the user could choose between three tabs: Sensors tab, History tab and Settings tab. By default, the Sensors tab is selected and presents a list view with the available WSNs for the current user. On this screen, if the user selects one of the available WSNs, a new list view is loaded with the mote names that belong to the selected network. When a unique mote is selected, the main data visualization screen is presented under the tab bar as shown in Fig. 6.



Figure 5. Login screen.



Figure 6. Data visualization screen.

This screen displays the latest sensor readings of the selected mote in both numerical and graphical modes. Following Fig. 6, “1” indicates the selected Sensors tab, while “2” points out the numerical data presentation where four sensor parameters are presented as well as each parameter unit. If more than four sensor readings are available for each mote, the user can select the four sensor readings to present simultaneously on the screen by using the button indicated by “3”. At the bottom of the screen, sensed data is presented graphically as indicated by “5”. Using the button pointed out by “4”, the user may choose the parameter to visualize in the graphical representation. These parameter readings are presented as a line graph that is updated as new values are received by the mobile device. The graph line represents sensed values over time and the y-axis scale is constantly adjusted in order to center vertically the line graph.

If the user switches to the History tab, a new screen is shown, which presents an interface that allows the user to choose a time interval as indicated by Fig. 7. When the user defines a time interval, a full-screen line graph is displayed in landscape orientation with all the sensor readings for the current parameter in that time interval. The chosen time interval may be days, hours or only a few minutes. Following Fig. 7, “1” indicates the selected History tab. Labels “2” and “3” represent the buttons that are used to define the limits of the time interval while “4” points out the button to draw the line graph.

The default application settings may be changed in the Settings tab as shown in Fig. 8. The settings menu provides the following application options: enable or disable data updates; define the updates frequency; and enable or disable push notifications. Following Fig. 8, label “1” points out the option used to enable or disable data



Figure 7. Historic data screen.

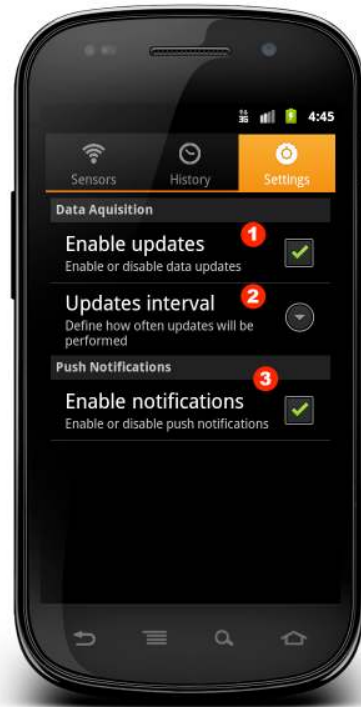


Figure 8. Settings screen.

updates. This option is enabled by default and is used to reduce battery consumption and data traffic. The frequency of data updates can be defined from 1 second to 1 minute by using the option indicated by “2”. When the data updates are disabled, this option is blocked automatically. Label “3” points out the option that allows the user to enable or disable push notifications.

6. Performance Evaluation and Demonstration

In order to evaluate and demonstrate the architecture and mobile application, a 6LoWPAN wireless sensor network laboratory testbed was constructed. In the design of the 6LoWPAN WSN, several Telos B motes running the TinyOS operating system

were used [30]. This network may be seen in Fig. 9. These motes communicate through IEEE 802.15.4 and the 6LoWPAN protocol stack is provided by the TinyOS Blip 1.0 implementation. The motes are capable of sensing air temperature and humidity, luminosity and battery voltage readings. A 6LoWPAN gateway is used to provide IPv6 end-to-end connectivity between the sensor network and the Internet. The 6LoWPAN gateway runs on Ubuntu 10.0.4 and it has multiple communication interfaces technologies, including IEEE 802.15.4, Ethernet and IEEE 802.11a/b/g. To implement the IEEE 802.15.4 interface in the gateway device, a TelosB mote connected to an USB port was used. An Intel desktop board D945GCLF with a 1.6 Ghz Intel Atom processor has been used to be the motherboard of the gateway. The application IP-driver compliant with RFC 4944, provided by TinyOS 2.1, acts as the 6LoWPAN adaptation layer in the gateway.



Figure 9. 6LoWPAN wireless sensor network laboratory testbed.

The 6LoWPAN gateway is also responsible for sending ICMPv6 router advertisement messages to announce the IPv6 prefix and the default gateway address to all sensor nodes.

The smartphone used to evaluate the proposed architecture and mobile application was the Samsung Galaxy S, running Android 2.3 with a 1.0 GHz CPU and a Li-Ion 1500mAh battery. All the tests were performed over an IEEE 802.11g connection with Internet access. During the experiments, the phone only used essential core services, the Wi-Fi adapter and the constructed application in foreground.

Measuring energy consumption on mobile devices is not easy. There are several factors that influence the energy consumption in a mobile device, specifically a smartphone. The energy consumption is different from device to device and it also depends of the operating system version and network specifications. For each experiment, the Android battery manager were used to check the current battery level that runs from 100% when fully charged to 0%.

Table I shows the relation between the amount of downloaded data, the battery consumption and the missed updates for a varying polling rate with a fixed update rate. The update rate of the wireless sensor network values was fixed in 5 seconds for testing purposes while the polling rate of the Android application varies between 3, 10 and 30 seconds. Furthermore, the same experiment was conducted with push notifications and also with the smartphone in idle for comparison purposes. All the tests were performed during 1 (one) hour of monitoring for each polling rate.

As expected, the lower the polling rate, the lower the battery consumption of the smartphone, but more updates missed by the monitoring application. The experiments also showed that if the polling rate is lower than the update rate of the

Table I. Relation between the downloaded data, battery consumption, and missed updates with changing polling rates and push notifications.

Polling Rate	3 sec	10 sec	30 sec	Push	Idle
Downloaded data/hour	501 KB	149 KB	49 KB	301 KB	_
Battery consumption/hour	19 %	13 %	9 %	4 %	2 %
Missed updates	0 %	50,4 %	83,6 %	0 %	_

WSN, none of the updates is missed but the amount of downloaded data is very high resulting in higher energy consumption. On the other hand, if the polling rate is too high, the amount of downloaded data and the battery consumption are reduced but the percentage of missed updates is also high. When the push notification system is used, the energy costs are significantly reduced and none of the updates are missed by the application. If push notifications are enabled, the mobile application is not constantly sending requests to the server and checking if there are any updates resulting in significant energy saving. As a result, ubiquitous wireless sensor networks monitoring is much more energy efficient when push technologies are used on mobile client applications.

The end-to-end connectivity between the smartphone and the 6LoWPAN WSN was also tested. Real-time temperature readings were collected and presented successfully in the mobile application.

7. Conclusion

This paper proposed a ubiquitous wireless sensor networks monitoring solution allowing users to receive latest sensor readings as well as historical measures on their

smartphones. The architecture was designed to be modular and was constructed based on open standards to ensure scalability and reusability. Since it is based on REST interfaces and XML/JSON messaging, the architecture is platform independent and supported in the majority of current mobile devices.

A push notification system was constructed specifically for mobile devices and is able to send push messages to smartphones if a sensor reading overcomes a given threshold. The smartphone application is also able to access real-time data over the Internet through a gateway software application. The proposed architecture was evaluated and demonstrated using a real wireless sensor testbed and an Android mobile application. The experiments showed that the solution is reliable and the push notification system has a significant impact on smartphone's energy savings.

As future work, the proposed solution may be tested in a real environment, deploying the wireless sensor network testbed outside the laboratory. In an outdoor environment, factors such as energy management, security and weather conditions should be considered. Furthermore, the mobile application could be extended to other mobile platforms such as the iPhone and Windows Phone.

Acknowledgements

This work has been partially supported by the *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Portugal, by National Funding from the FCT – *Fundação para a Ciência e Tecnologia* through the Pest-OE/EEI/LA0008/2011, and by the AAL4ALL (Ambient Assisted Living for All), project co-financed by COMPETE under FEDER via QREN Programme.

References

- [1] A. Hornsby, P. Belimpasakis, I. Defee, *XMPP-based wireless sensor network and its integration into the extended home environment*, in ISCE, IEEE 13th International Symposium on Consumer Electronics, May 25-28, 2009.
- [2] A. Ludovici, A. Calveras and J. Casademont, *Forwarding Techniques for IP Fragmented Packets in a Real 6LoWPAN Network*, *Sensors* **11**(1), 2011, pp. 992-1008.
- [3] A. Munawar, A. Masood and F. Bangash, *Open sensor platform: Integration of sensors and mobile phones*, in IBCAST, International Bhurban Conference on Applied Sciences and Technology, January 9-12, 2012.
- [4] Adobe Flex Framework, <http://www.adobe.com/devnet/flex.html>.
- [5] Android Cloud To Device Messaging.
<http://developers.google.com/android/c2dm/>. Accessed Jan 2012.
- [6] Android User Interface Guidelines,
[http://http://developer.android.com/guide/practices/ui_guidelines/index.html](http://developer.android.com/guide/practices/ui_guidelines/index.html).
Accessed Jan 2012.
- [7] B. da Silva Campos, J.J.P.C. Rodrigues, L.M.L. Oliveira, L.D.P. Mendes, E.F. Nakamura and C.M.S. Figueiredo, *Design and construction of a wireless sensor and actuator network gateway based on 6LoWPAN*, in EUROCON, International Conference on Computer as a Tool, April 27-29, 2011.
- [8] C. Alcaraz, P. Najera, J. Lopez and R. Roman, *Wireless Sensor Networks and the Internet of Things: Do We Need a Complete Integration?*, 1er International Workshop on the Security of The Internet of Things, 2010.
- [9] D.S. Tudose, A. Voinescu, M. Petrareanu, A. Bucur, D. Loghin, A. Bostan, M. Tapus, *Home automation design using 6LoWPAN wireless sensor networks*, in DCOSS, International Conference on Distributed Computing in Sensor Systems and Workshops, June 27-29, 2011.
- [10] F. Belqasmi, R. Glitho and F. Chunyan, *RESTful web services for service provisioning in next-generation networks: a survey*, *IEEE Communications Magazine*, **49**(12), 2011, pp. 66-73.
- [11] I.F. Akyildiz, Su Weilian, Y. Sankarasubramaniam and E. Cayirci, *A survey on sensor networks*, *IEEE Communications Magazine* **40**, 2002, pp. 102-114.
- [12] J. M. Corchado, J. Bajo, D.I. Tapia and A. Abraham, *Using Heterogeneous Wireless Sensor Networks in a Telemonitoring System for Healthcare*, *IEEE Transactions on Information Technology in Biomedicine* **14**(2) 2010, pp.234-240.

- [13] Java API for Restful Services. <http://jax-rs-spec.java.net/>. Accessed Nov 2011.
- [14] Jersey Open-source Framework. <http://jersey.java.net/>. Accessed Nov 2011.
- [15] L. Atzori, A. Iera and G. Morabito, The Internet of Things: A survey, *Computer Networks* **54**(15), Elsevier, 2010, pp. 2787-2805.
- [16] L. Herrera, B. Mink, S. Sukittanon, *Integrated personal mobile devices to wireless weather sensing network*, Proceedings of the IEEE SoutheastCon, March 18-21, 2010.
- [17] L. M. L. Oliveira and J. J. P. C. Rodrigues, *Wireless Sensor Networks: a Survey on Environmental Monitoring*, *Journal of Communications* **6**(2), 2011, pp. 143-151.
- [18] L. M. L. Oliveira, A. F. de Sousa and J. J. P. C. Rodrigues, *Routing and mobility approaches in IPv6 over LoWPAN mesh networks*, *International Journal of Communication Systems* **24**(11), Wiley, 2011, pp. 1445–1466.
- [19] L. Mottola and G. P. Picco, *Programming wireless sensor networks: Fundamental concepts and state of the art* **43**(3), *ACM Computing Surveys*, 2011.
- [20] M. Cardei, A. Marcus, I. Cardei, T. Tavitlov, *Web-based heterogeneous WSN integration using pervasive communication*, in IPCCC, IEEE 30th International Performance Computing and Communications Conference, November 17-19, 2011.
- [21] M. Nottingham and R. Sayre, *The Atom Syndication Format*, RFC 4287, December 2005.
- [22] MySQL Database Management System. <http://dev.mysql.com/doc/>. Accessed Oct 2011.
- [23] N. Aslam, W. Phillips, W. Robertson and S. Sivakumar, *A multi-criterion optimization technique for energy efficient cluster formation in wireless sensor networks*, *Information Fusion* **12**(3), Springer, 2011, pp. 202-212.
- [24] N. Moreira, M. Venda, C. Silva, L. Marcelino, A. Pereira, *@Sensor - Mobile application to monitor a WSN*, in CISTI, 6th Iberian Conference on Information Systems and Technologies, June 15-18, 2011.
- [25] P. Sanyal, S. Das, S.S. Bhunia, S. Roy and N. Mukherjee, *An experience of implementing IPv6 based data retrieval system for Wireless Sensor Networks*, on RACSS, International Conference on Recent Advances in Computing and Software Systems, April 25-27, 2012.
- [26] P. Vajsar, L. Rucka, *Monitoring and management system for wireless sensor networks*, 34th International Conference on Telecommunications and Signal Processing, August 18-20, 2011.

- [27] P. Wang, Z. Sun, M. C. Vuran, M. A. Al-Rodhaan, A. M. Al-Dhelaan and I. F. Akyildiz, *On network connectivity of wireless sensor networks for sandstorm monitoring*, Computer Networks **55**(5), Elsevier, 2011, pp. 1150-1157.
- [28] R. Kemp, N. Palmer, T. Kielmann and H. Bal, *Energy Efficient Information Monitoring Applications on Smartphones through Communication Offloading*, Mobile Computing, Applications, and Services, **95**(2), Springer, 2012, pp. 60-79.
- [29] R. T. Fielding, *REST: architectural styles and the design of network-based software architectures*, Doctoral dissertation, University of California, Irvine, 2000.
- [30] Tiny OS Documentation Wiki, <http://docs.tinyos.net/tinywiki/index.php/>. Accessed Feb 2012.
- [31] V. Kumar and S. Tiwari, *Routing in IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN): A Survey*, Journal of Computer Networks and Communications, 2012.
- [32] Z. Li, Y. Liu, M. Li, J. Wang and Z. Cao, *Exploiting Ubiquitous Data Collection for Mobile Users in Wireless Sensor Networks*, IEEE Transactions on Parallel and Distributed Systems, Issue 99, 2012.

Author Biographies

André Gaudêncio F. Elias is an MSc student of Informatics Engineering at the University of Beira Interior under the supervision of Professor Joel Rodrigues and Professor Bruno Zarpelão. He received his 3-year BS degree in Informatics Engineering from the University of Beira Interior, in 2010. In 2011, he studied in the University of Campinas, Brazil, during one MSc semester. He is an MSc student member of the Next Generation Networks and Applications Group (NetGNA) at Instituto de Telecomunicações, Portugal. His main research areas include wireless sensor networks, and mobile and ubiquitous computing.

Joel J. P. C. Rodrigues (joeljr@ieee.org) is a professor at the University of Beira Interior (UBI), Covilhã, Portugal, and researcher at the *Instituto de Telecomunicações*, Portugal. He received a PhD degree in informatics engineering, an MSc degree from the University of Beira Interior, and a five-year BSc degree (licentiate) in informatics engineering from the University of Coimbra, Portugal. He is the Director of the Master degree in Informatics Engineering at UBI. He is the leader of NetGNA Research Group (<http://netgna.it.ubi.pt>), the Vice-chair of the IEEE ComSoc Technical Committee on Communications Software, the Vice-Chair

of the IEEE ComSoc Technical Committee on eHealth, and Member Representative of the IEEE Communications Society on the IEEE Biometrics Council. He is the editor-in-chief of the International Journal on E-Health and Medical Communications, the editor-in-chief of the Recent Patents on Telecommunications, and editorial board member of several international journals. He has been general chair and TPC Chair of many international conferences. He is a member of many international TPCs and participated in several international conferences organization. He has authored or coauthored over 220 papers in refereed international journals and conferences, a book, and 2 patents. He had been awarded the Outstanding Leadership Award of IEEE GLOBECOM 2010 as CSSMA Symposium Co-Chair and several best papers awards. Prof. Rodrigues is a licensed professional engineer (as senior member), member of the Internet Society, an IARIA fellow, and a senior member of ACM and IEEE.

Luís M. Oliveira (loliveira@ipt.pt) is a PhD student of Informatics Engineering at the University of Beira Interior, Covilhã, Portugal, under supervision of Professors Joel Rodrigues and Amaro de Sousa. He received his 5-year BS degree (licentiate) in Electronics in 1998 and the MSc degree in Electronics and Telecommunications Engineering in 2004, both from the University of Aveiro. He also teaches at the Polytechnic Institute of Tomar, Portugal. He is a PhD student member of the Institute of Telecommunications, Portugal. His research interests include routing on wireless sensor mesh networks. He has authored or co-authored over ten papers in international refereed journals and conferences.

Bruno B. Zarpelão received his B.S. degree in Computer Science from State University of Londrina, Brazil, and the Ph.D. degree in Electrical Engineering from University of Campinas, Brazil. He is currently a professor at the *Pontifícia Universidade Católica do Paraná* (PUC-PR), Londrina, Brazil and an associate researcher at the Next Generation Networks and Applications Group (NetGNA), University of Beira Interior, Covilhã, Portugal. His research interests include Smart Cities, eGov, Open Access MAN, Communication Network Management and Information Security.

4. Conclusions and Future Work

This chapter presents the main conclusions of this dissertation and points further research directions. The main objective of this dissertation was to design and development of an architecture to enable mobile devices, specifically smartphones, to remotely monitor wireless sensor networks in ubiquitous environments. This was accomplished with the construction of a Web service, a relational database, and an Android mobile application, used as a prototype to demonstrate, evaluate, and validate the proposed solution. Besides, end-to-end connectivity was established between the smartphone and the WSN through a gateway. To enable the user for being alerted when significant changes occur on the WSN, a push notification system was also developed and integrated in the mobile application. The constructed mobile application is able to present real-time and historical sensor readings in a simple and meaningful way as well as to receive push notifications. Therefore, all the dissertation objectives were successfully accomplished.

After a brief introduction regarding wireless sensor networks and mobile computing, the objectives and the main contributions were presented. Chapter 2 presents the first contributions and describes the construction of a ubiquitous model for wireless sensor networks monitoring. Chapter 3 describes, in detail, the constructed architecture and the mobile application. It also introduces a push notification system and an end-to-end connectivity functionality that allow real-time monitoring. Also, performance evaluation and validation of the solution are presented.

The solution was demonstrated and evaluated through a real wireless sensor network testbed and the results show that solution is reliable and the push notification system has a significant impact on smartphones energy savings.

To conclude this document, it remains to point future research directions that result from this work. As future work, the proposed solution may be experimented in a real environment, deploying a bigger wireless sensor network outside the laboratory. In outdoor environment, other factors, such as energy management, security, and weather conditions should be considered. Furthermore, the mobile application could be extended to other mobile platforms such as the iPhone and Windows Phone.

