2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing

A Ubiquitous Model for Wireless Sensor Networks Monitoring

André G. F. Elias[1], Joel J. P. C. Rodrigues[1], Luís M. L Oliveira[1], Bruno B. Zarpelão[2],

[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.

978-0-7695-4684-1/12 $26.00 © 2012 IEEE
DOI 10.1109/IMIS.2012.33

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 trough 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.

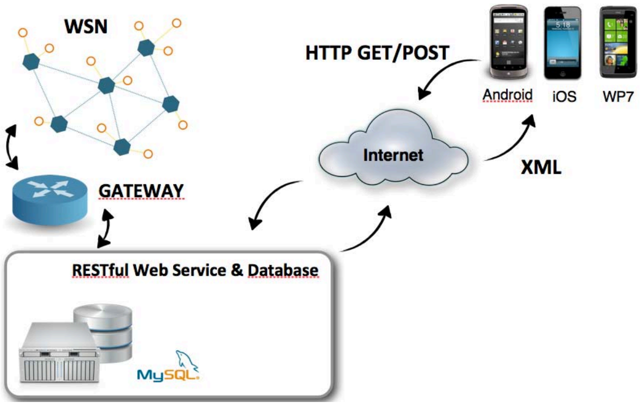

Figure 1. Illustration of the system architecture diagram.

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.

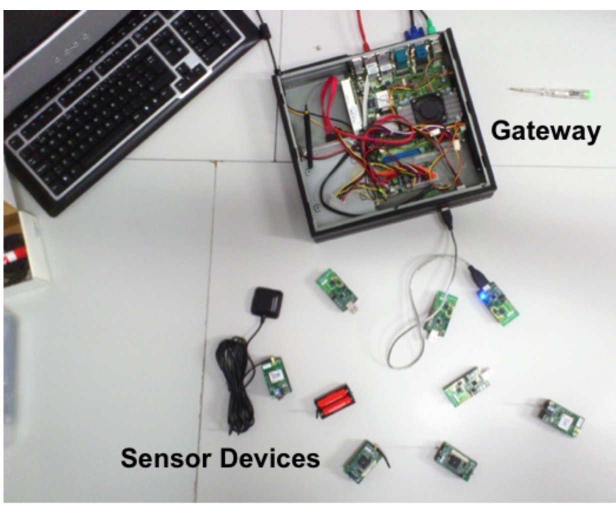

Figure 2. Photo of the used WSN laboratory testbed with sensors and the network gateway.

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.

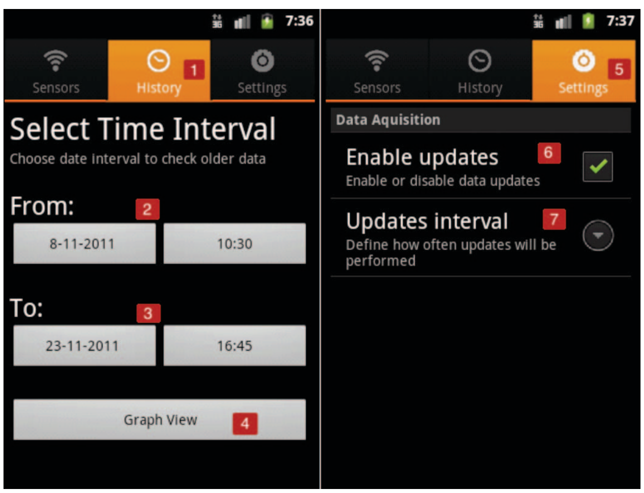

Figure 4. History and Settings screens.

VI. Conclusion 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.

Acknowlegments

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 systembased 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