A MONITORING SYSTEM FOR MULTIPLE RESOURCES SUPPORTING SMART HOUSE AND BUILDING AUTOMATION

Marcello Pietri and Riccardo Lancellotti
University of Modena and Reggio Emilia

ABSTRACT

In the last few years, a growing interest towards reducing resource consumption and carbon footprint has been the main driver for the development of intelligent systems for smart houses and buildings.
Existing solutions are limited to monitor and optimize the usage of each single utility or resource, such as electric power, water and gas. However, higher savings can be achieved by monitoring and collecting data from multiple utilities and sensors in combination at the same time, instead of focusing on the optimization of the single factor.
We present a novel solution to support integrated monitoring and data collection in a smart house with the objective to reduce utility cost and toxic byproducts. We discuss the requirements for the metering system and we propose an architectural solution satisfying these requirements. Furthermore, we describe a system prototype implementing the proposed architectural solution. The experiments carried out on the prototype show that our proposal is a viable option to support the monitoring of multiple resources interacting with heterogeneous sensors and external interfaces.

KEYWORDS

Smart House, Automatic Meter Reading (AMR), ZigBee, Wireless M-Bus, Building automation.

1. INTRODUCTION

New technologies such as ultra low power microcontrollers with radio frequency communications and high precision low power sensors are the foundation for automatic monitoring systems that support the reduction of environment footprint and costs related to resource utilization in future smart houses and buildings. Currently available solutions for monitoring offer limited features, as they focus mainly on a single type of resource (e.g., electrical power) or they take into account only a limited type of sensors and applications (e.g., Zigbee based power meters for interaction with a specific electrical company) (Xu, et al., 2008; Thepphaeng & Pirak, 2011; Zhu, et al., 2006; Yanfei, et al., 2009; Shan, et al., 2008).

This approach places unacceptable constraints to both users and resource providers, as it requires a separate monitoring system for each resource and for each provider, leading to higher costs. Furthermore, this approach to monitor separately each resource is hindering the development of integrated solutions, which in fact will help to better understand consumption patterns, leading to service improvement and logistics optimization simply by analyzing several resources and factors at once.

Available solutions can only support analysis based on a single resource. For example that systems could report "Based on your energy consumption patterns, you could save 360€ per year by upgrading to a more efficient electric water heating" (Froehlich, et al., 2011). Monitoring all the resources can support analyses that provide the users with a full consciousness about the status of all resources consumed by each single object. The new system could report, for example, "Based on your energy, gas and water consumption patterns you could save: 360€ per year by upgrading to a more efficient electric water heating; 470€ per year by switching to a gas water heating, 980€ per year by switching to a solar one. Using the automatic system to manage the boiler, by heating only when is really necessary you could save another 35% of gas or electricity"

Our solution is the proposal of a novel system for the automated monitoring of multiple resources (such as gas, water and electricity). We present the architecture of the proposed system and we provide a detailed view of a prototype implementation of a gateway that can collect data from heterogeneous sensors and supports interaction with multiple service providers for metering purposes.

Preliminary experiments on the prototype monitoring system demonstrate its viability as a metering solution. In particular we show how the prototype can support heterogeneous sensors over a building. System scalability is evaluated with respect to the number of sensors and data collection process is discussed with emphasis on network failure recovery and safe operation during power outage.

The paper is organized as follows: Sections 2 and 3 describe the system requirements and proposed prototype implementation, Section 4 illustrates the preliminary tests, Section 5 reports state-of-the-art solutions, Section 6 concludes with final remarks and introduces the discussion for the future studies.

2. MONITORING SYSTEM REQUIREMENTS

We now describe our proposal to support the monitoring of multiple resources through heterogeneous sensors and meters. Our goal is to overcome the main limitation of existing solution for automated monitoring where data collection is limited to homogeneous devices.

We focus on an architecture consisting of a central gateway that collects data from the sensors and meters and sends them to remote nodes. This is a common approach for automatic meter reading systems. However, we identify three areas where improvements with respect to the available solutions are required. We motivate the requirements by pointing out potential benefits for both companies that serve resources (such as gas, water and electricity) and for end users. Specifically, we summarize the requirements motivating our proposal in three items:

1. capability to support heterogeneous communication systems;
2. availability;
3. scalability.

2.1 Capability to support Heterogeneous Communication Systems

Heterogeneity in communication systems can occur both at the level of Wireless Sensor Network (WSN) and at the level of gateway interface with external systems. With respect to the WSNs, many resource providers offer multiple resources (for example PG&E, AGL, Enel, … that provide both electricity and gas). Deploying a different gateway for any group of meters or sensors can be very expensive. By monitoring multiple different resources at the same time, the proposed architecture can reduce to one the number of required appliances. As a consequence, the staff involved in system planning, maintenance, and other features, must only have a detailed knowledge about one gateway system. The meters and the sensors can be various and built by different firms that adopt the ZigBee or Wireless M-Bus protocols.

The gateway must be connected to the network via different interfaces, such as Ethernet or Wi-Fi that may rely on the user DSL connection, or can be equipped with HSDPA to connect to a mobile network. If both connectivity are supported, the gateway can autonomously decide if it is better exploit a free DSL connection, that may be temporarily unavailable (for example if the user turns off its modem), or if there is the need to instantiate a connection to the mobile phone network. Using these interfaces, the proposed architecture can also send data to the cloud data centers, e.g. to Pachube (Haque, et al., 2011). This on-line database allows the developers of resource providers (or to third parties) to feed sensor data, and supports the deployment of software for analysis and of value added applications.

2.2 Availability

For availability we consider that the gateway must be able to receive all the data from the devices, store them, and support basic functions of data filtering, aggregation and reporting for the user. To support this requirement we need an adequate level of performance in terms of CPU, RAM and storage. Furthermore, as the data collection process must be carried out also in the case of electric power outage, we need to consider a backup battery and we must ensure low power consumption for every device involved in the data collection process, to ensure a long battery lifetime.

The local storage of the devices data is required to cope with a possible unavailability of the connection with the external data collectors. To this aim, besides the ability to buffer data and synchronize the storage with a remote repository, data aggregation and filtering should be possible locally in order to provide basic services for the users without a remote third-party involvement. For example the users must be able to consult all the data about the consumptions and the rooms' status by interacting only with the gateway, possibly through a friendly user interface.

2.3 Scalability

The need for scalability is motivated considering that the proposed architecture should be used in different scenarios ranging from a large building to a small house. While other solutions follow different system setups to cope with scenarios of different size (Lam, et al., 2008), we aim to provide a more flexible and scalable solution. In other words, by adopting the proposed architecture the same gateway can be used in all contexts only by removing or adding peripheral devices. The number of sensors supported by the gateway can range from a few units up to hundreds of devices. Furthermore, components providing additional features (such as a touchscreen-based user interface) can be added or removed in order to satisfy the costs and the energy usage constraints.

3. PROTOTYPE IMPLEMENTATION

The developed prototype is based on the requirements described in Section 2. It aims to support heterogeneous communication systems and to ensure the system availability and scalability.

3.1 The Architecture

In order to satisfy the described requirements, we propose the architecture in Figure 1. The automated monitoring system is composed by three main parts: the Device Sphere (DEVS), the Gateway Management System (GWMS), and the User Sphere (USRS). With the term "devices sphere" we intend all the meters (e.g., gas, water and electricity) and all the sensors, (e.g., luminance, flooding, CO_2 and temperature) that the GWMS is able to support. With term "gateway management system" we intend all the components that are able to collect the data from the DEVS, transform this data in an intelligible format and provide it to the USRS. With terms "user sphere" we intend all the software (such as analysis software, data reporting interfaces and applications) that a user, human or not, can use to interact with the GWMS.

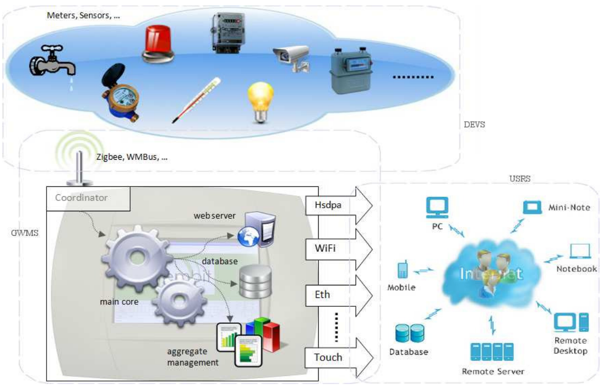

Figure 1. System architecture overview

The GWMS and the DEVS are composed by two layer, hardware and software. The hardware and the software of the DEVS can be very heterogeneous and depends on each single constructor. The software layer is usually a proprietary firmware compliant with a standard protocol, that the coordinators should support.

About the DEVS, we decide to support ZigBee and WMBus protocols. The ZigBee is currently one of the most used standards for LoWPANs. It supports many devices, as light sensors, electrical meters, and other consumer and industrial equipment that requires short-range wireless transfer of data at relatively low rates. The WMBus is radio variant of M-Bus European standard (EN 13757-2/3). These two protocols allow the GWMS to monitor multiple different devices and resources, making easier the use of the same architecture in various contests, such as building automation and smart house.

The proposal is an embedded machine able to support heterogeneous communication systems. This system will be able to support from few to hundreds of DEVS, also during electrical power outages. The system will easily support the selected interfaces and peripherals in order to provide services to the USRS. The network interfaces (as Wi-Fi, Ethernet, …) support remote monitoring by USRS, potentially interacting with cloud-based platforms, while the peripherals allows the persons to interact directly with the DEVS.

3.2 Hardware

The hardware used to build the prototype can be divided in two fields, the GWMS (components and interfaces) and the DEVS. The interfaces of GWMS (such as usb keys, Ethernet and touchscreen) and DEVS can be very heterogeneous and its hardware depends on each single device. On the contrary the GWMS components must be known a priori; it is composed by three main components: the main board, the ZigBee and the wireless modbus coordinators.

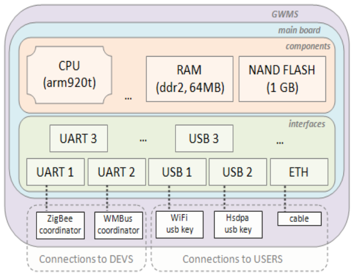

Figure 2. Hardware schema

The Figure 2 shows the hardware created and used for the main board of GWMS (layer in Figure 1). This is an embedded platform mounting an ARM920T CPU, 64Mb of RAM and 1 GB of flash. It was built in order to respect low power and low cost features. This solution is currently equipped with many interfaces as Ethernet, Wi-Fi (WPA2, WPA, and WEP), HSDPA (and UMTS (2100 MHz), EDGE, GPRS, GSM (850/900/1800 MHz), SMS enable), UART/RS232, I2C and SPI. Many peripherals are also available, for example touchscreens (3.5", 4.3" or 7"), USB (keyboards, mices, data storages, …) and SD cards.

The Zigbee Coordinator is an OEM wireless module developed by Embit (2011) for LR-WPAN applications. The module combines high performance to small dimensions and low costs, providing the system integrator a simple and easy way to add IEEE 802.15.4 / 6LoWPAN / ZigBee / RF4-CE low range wireless connectivity and multi-hop networking into existing products. EMB-Z1322PA is configured as an embedded micro system or simple data modem for low power applications in the 2.4 GHz ISM band. It is based on a Freescale™ MC13224V single chip device which is an ARM7 32 bit controller with 128 Kbytes Flash memory, 96 Kbytes of RAM and 80 Kbytes of ROM. The device includes a hardware accelerator for the MAC IEEE802.15.4, AES security and a 2.4 GHz transceiver.

The Wireless M-Bus coordinator combines high performance to small dimensions and low costs, providing the system integrator a simple and easy way to add WMBus wireless connectivity and multi-hop networking into existing products (*Embit*, 2011). It is configured as an embedded micro system or simple data modem for low power applications in the 169 MHz or 868/915 MHz ISM band. It is based on Texas Instruments CC1120 coupled with an MSP430 core 5 MCU equipped with up to 128 Kbyte of Flash memory and up to 10 Kbyte of RAM memory.

3.3 Software

The embedded software of the coordinators is a firmware developed by the constructor. These coordinators communicate with the main board through a RS232 and/or a USB interface. The protocols are specific for each different type of coordinator. About the software of main board, is possible to divide it in two groups, the Operative System (OS) and the applications. The operating system kernel is the Linux kernel version 3.0.4 and main userspace software is based on Busybox 1.19.2.

Running on the OS, it is possible to distinguish two types of software: third-party utilities and applications developed by us. In the first group, we have a HTTP server, a database and an SSH server. By comparing different small http servers (*Web Server Comparisons*, 1998), we choose the Thttpd server as it provides the best size-efficiency trade-off. An important feature embedded systems with low computational power is the bandwidth throttling feature that prevents the connection from becoming saturated. The selected SSH (and SFTP, SCP) server is Dropbear, while two different databases are provided: RRDtool (round-robin database tool) and SQLite. RRDtool is the open source industry standard, high performance data logging and graphing system for time series data. RRDtool can be easily integrated in shell scripts, Perl, Python, Ruby, lua or tcl applications. The data are stored in a round-robin database (circular buffer). Hence the system storage footprint remains constant over time. It also includes tools to extract RRD data in a graphical format. SQLite is a software library that implements a self-contained, serverless, transactional SQL database engine.

Three main applications internally developed aim to manage the system, namely the MxZC, the MxWMB and the GWCore. MxZC and MxWMB are the two software that interact with the ZigBee coordinator and the Wireless M-Bus coordinator. The MxZC includes also part of the ZigBee Cluster Library (ZCL) that is developed as an independent library. These two programs provide an interface between coordinators and

GWCore, that becomes able to communicate with the DEVS. The GWCore takes in charge all the activities about data storage, elaboration, analysis and delivery. To do this, it uses also the SSH2, HTTP and HTTPS protocols, and the HTML, CSV, XML and JSON standard formats. The GWCore also have a graphic user interface (GUI) that is directly included in the software in order to obtain better performance. This feature could be enabled at compile-time, allowing to mount or not the touchscreen on GWMS. The GUI could be displayed by remote desktop using a VNC client, also in a secure mode (SSH tunneling).

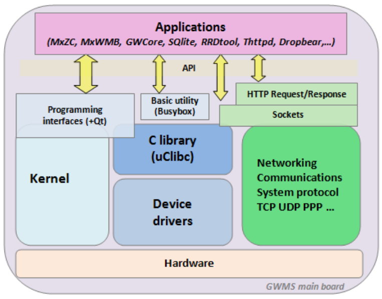

Figure 3. Software schema

4. PRELIMINARY TEST

We carry out a set of preliminary test to validate the proposed prototype, with the goal to demonstrate the viability of our solution for the automated monitoring of heterogeneous resources in multiple environments.

4.1 Experimental Setup

Recalling the main requirements of monitoring multiple resources guaranteeing availability and scalability of the data collection process, we carry out the following tests.

1. Measurement of networks coverage. This test aims to verify if the selected protocols could be used in the houses and also in public buildings with different types of sensors (using different network protocols). We expect that the one-hop coverage is at least: ten meters (a big room) for the ZigBee network; hundred meters (a building area) for the WMBus network.

2. Evaluation of GWMS energy consumption. The objective of this test is to prove the real energy consumption of the whole system, including some communication/user interfaces. Reduced energy consumption is a key feature in case of power outage as it increases the lifetime of batteries. In this test we don't take care of the DEVS' battery lifetime because: it depends on the individual manufacturers; the DEVS send reports to GWMS when the battery becomes low. In any case, according to manufacturer's tests, in the used DEVS the battery of sensors have a lifetime of ca. 2 years while the meters one of ca. 10 years. In case of electrical power outage the key-hardware in terms of consumption will be the GWMS. The expectation is a consumption at most of 5 Watt per hour (max 2,5Wh for an usb key and max 2,5Wh for the main board, coordinators and touchscreen).

3. Performance evaluation of OS and applications. The objective of these tests is the validation of requirements concerning real-time interactivity with USRS, remote administration, and support for multiple different resources. Furthermore, we validate the support of multiple concurrent services. What we expect is the GWMS response at most in 8 seconds for served Web pages and at most of 4 seconds for other services as GUI on touchscreen. These times are related to standard networks status and for one-USRS access (house scenario).

4. Evaluation of the maximum number of supported DEVS. The main objective of this test is the evaluation of the system scalability in extended scenarios such as public or apartment buildings. We expect that the GWMS, without touchscreen GUI, can support networks composed at least form 100 DEVS (e.g. 3 meters and 7 sensors for house, 10 houses). We suppose that these DEVS send the information at most every 30 seconds.

4.1.1 Networks Coverage

In the first test we measure the maximum distance for which the coordinators communicate with one of their device. During the test the antenna is located at one meter from ground level.

Erro! A origem da referência não foi encontrada. shows the maximum distance for different network technologies and scenarios. From our experiments the maximum network coverage is 55m for ZigBee, while for WMBus it is 480m.

ISBN: 978-972-8939-67-0 © 2012 IADIS

Table 1. Networks coverage

Network type	Coordinator position	Device position	Objects in the middle	Max distance
ZigBee	into a building	into same building	walls, doors, …	25 m
ZigBee	into a building	outside	buildings, car, walls, …	35 m
ZigBee	outside	outside	buildings, cars, …	55 m
Wireless M-Bus	into a building	into same building	walls, doors, …	190 m
Wireless M-Bus	into a building	outside	buildings, car, walls, …	280 m
Wireless M-Bus	outside	outside	buildings, cars, …	480 m

4.1.2 GWMS Energy Consumption

In this test we try the GWMS energy consumption. The GWMS runs with attached the HSDPA usb key and a 3,5" touchscreen, always on. The used battery is a rechargeable one of 20W (5V, 4A); the scenario's temperature between 20 and 28°C. During the various tests, the lifetime resulted approximately of 10 hours.

4.1.3 OS and Applications Performance

In these tests we report information about CPU and RAM usage obtained from Top utility program included in BusyBox. In the first test scenario we try the whole system without active devices in the networks, but with a fixed amount of stored data (ca. 30 DEVS). The tests were done during a user interaction (iter) or not (idle). The user act via touchscreen/VNC or Web server, depending on scenario. In fact, two different configurations were been tested: the first with or without GUI for touchscreen and VNC; the second with or without VNC server.

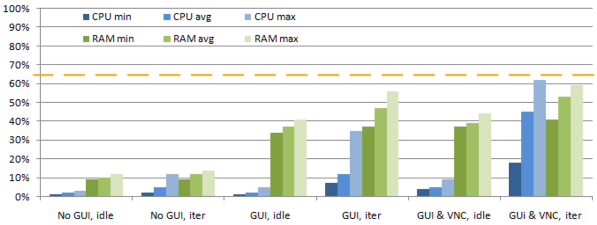

Figure 4. OS and applications resources usage

The GWMS without GUI consumes at most the 12% of CPU and less than 9MB of RAM (14%) even in interactive mode. Adding the GUI (touchscreen), the GWMS consumes from 1% to 35% of CPU, while the RAM usage increases approximately from 22 to 36MB. The max CPU and RAM usage happens during user interactions via VNC (GUI enable). However, in this case the max resources usage is at most the 62%.

In order to test the performance of GWMS, a stress test has been done. Some users, 1 by SSH (Dropbear; login and directory list), 3 by HTTP (Thttpd; GWMS management and DEVS data view), 1 by touchscreen/VNC (DEVS data view), interacted with 3 WMBus meters and 6 ZigBee sensors. The response time was measured into the applications code. The system serves all the users' requests at most in 3 seconds.

4.1.4 Supported DEVS Number

In this test we simulate a ZigBee network composed from many devices (light and temperature sensors, IAS devices, …) and one coordinator. The coordinator communicates to the GWMS the requests (joins, enrolls) and the data collected from the sensors (sent every 30 seconds). The GWMS stores the data, enqueues and manages all the requests and sends replies to the coordinator. The objective is to monitor how many devices are supported before resource saturation in the GWMS (utilization=100%).

Table 2. Application CPU usage

Device number	CPU avg (adding)	steady time (adding)	CPU avg (added)
25	37%	0s	9%
50	52%	0s	22%
75	79%	0s	36%
100	91%	0s	47%
125	100%	2s	60%
150	100%	5s	74%
175	100%	16s	85%
200	100%	27s	97%
225	100%	>60s	100%

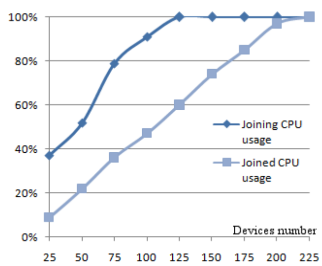

Figure 5. Application CPU usage

The results show that the CPU is the bottleneck resource. Table 2 and Figure 5 show the CPU usage during and after the join (and enroll for some devices) phase. In particular the 2nd and 3rd columns data are related to "during the join phase" (adding), while the 4th column is related to "after the join phase" (added). During the joining the CPU supports up to 100 DEVS before utilization reaches the 100% and up to 200 DEVS with a little time-wait. After the join phase the CPU can manage up to 200 DEVS.

4.2 Result Discussion

Starting from the expectations presented in Section 4.1, in the networks range test (Section 4.1.1) we see that all the objectives has been reached. The coverage of the ZigBee network is able to serve the house scenario, as well as the Wireless M-Bus network coverage is able to serve a building area scenario. In this test we try a one-hop ZigBee network, while is also possible to set up a multi-hop one.

About the energy consumption test presented in Section 4.1.2, the result proves that GWMS has lower energy consumption. This feature assures the autonomy of 10 hours in case of an electrical system failure. This low power consumption is also appreciated from a cost-of-energy point of view.

The results about test in Section 4.1.3, confirm that the GWMS has a real-time interactivity with USRS, is remotely administrable, and supports multiple different resources monitoring. In addiction the results show that the GWMS is also able to support a multi-user access.

In the last test (Section 4.1.4) we show that the supported network can be very large. In details, we find out that GWMS is able to support 200 DEVS which send data every 30 seconds, instead of the expected 100 DEVS (Thepphaeng and Pirak, 2011). With these results we can assure that the developed architecture can be employed in extended scenario as public or apartment buildings.

5. RELATED WORK

Automated resource monitoring is a critical research topic, as testified by the amount of research carried out both by the academia and by private companies. In particular, the research focuses on two main topics. First, the integration of solutions at the level of wireless sensor network to collect data related to the utilization of resources, second the development of complete solutions for resource monitoring.

With respect to the literature concerning wireless sensor networks for resource monitoring several proposals are available (McLauchlan & Bessis, 2011). ZigBee is the most popular protocol for wireless sensor networks (Xu, et al., 2008; Thepphaeng & Pirak, 2011; Zhu, et al., 2006; Yanfei, et al., 2009; Shan, et al., 2008). The ZigBee topology can range to flat wireless mesh (Zhu, et al., 2006) to centralized systems (Xu, et al., 2008). However, in these studies metering is limited to a single resource because only one type of meter is considered and a single protocol for interaction with sensors is taken in account. Furthermore, these studies are focused on wireless protocol communication detail and do not address the issues of supporting high-level analysis and aggregation of these data.

Another area of research is that of meter reading techniques. While the main goal of Jiang (2010) and Lam (2008) and the most part of the industry is to find the best solutions to transmit the meters data to the companies, the aim of Froehlich (2011) is to shown this data in a useful way to the single users. Other studies taking into account different resources are HydroSense (Froehlich, et al., 2009), to monitor water consumption and detect leaks, and GasSense (Cohn, et al., 2010) as well as Kempton and Layne (1994) focusing on gas consumption. Other analyses related to the issue of metering techniques are found in the Italian Telegestore project (Rogai, 2007) and in the list of vendors that provide a Smart Metering solution (Engelen & Collins, 2010; *ZigBee Alliance*, 2002).

All these efforts aim to monitor a single resource, electricity being the most common example. However, monitoring a single resource inherently limits the ability to support advanced analyses that can provide a significant reduction in resource utilization and consequent carbon footprint. We consider that providing a metering solution that can correlate multiple data sources can lead to a better understanding of user behavior and can improve the effectiveness of the proposed solution to optimize resource utilization.

6. CONCLUSION AND FUTURE WORK

The currently available solutions for automated metering of resources cannot integrate multiple heterogeneous sensors. However, such features could reduce the cost for data collection and could support advanced analysis aiming to reduce carbon footprint and user expenses.

We propose a novel architecture for automated metering which collects data from multiple sensors and resources, and regulates the utility for the maximum benefit of the consumer, for the minimum toxic emissions and for maximizing the profits of the utility provider. Our design explicitly aims to provide the following characteristics: first, the ability to support several communication protocols for both interaction with sensors and data delivery to remote centers where advanced analyses and value-added services can be provided. Second, availability of the data collection service even in the case of network failure and power outage. Third, scalability up to hundreds of sensors and meters to cover large scale deployment scenarios.

We present a prototype implementing the architectural design and we evaluate its ability to satisfy the design criteria. Our preliminary experiments demonstrate the ability of the prototype system to interact with up to 200 sensors scattered over a large area (up to 480m) and we demonstrate that the monitoring process can be carried out even in the case of power outages lasting up to 10 hours.

Future improvements of the prototype system include the development of applications (deployed locally or remotely) that can support in an intelligent system for disaggregate monitoring, including the ability to predict future resource utilization. This can support what-if analyses for devising utilization patterns, cost plans and hardware upgrades. An additional step is to allow some devices, such as water heating, central heating, air-conditioning, etc., to be automatically managed by GWMS. This last step enables the creation of fully automated smart houses that can provide high comfort at minimum cost.

REFERENCES

McLauchlan, N. and Bessis, N., 2011. Towards Remote Monitoring of Power Use: A Case for Smart Meters. *Sixth International Symposium on Parallel Computing in Electrical Engineering*. Luton, UK, pp. 133-138.

Engelen A. G. van, Collins J. S., 2010. Choices for Smart Grid Implementation. *43rd Hawaii International Conference on System Sciences*, pp. 1-8, Poipu, Kauai, Hawaii.

Jiang Y., 2010. Wireless Digital Gas Meter with Lower Power Consumption. *Fifth International Conference on Frontier of Computer Science and Technology*. Changchun, China, pp. 192-197.

Rogai S., 2007. Telegestore project progress & results. *IEEE ISPLC*. Pisa, Italy.

Xu C., et al., 2008. Automatic Electric Meter Reading System Based on ZigBee. *WiCOM/IEEE, International Conference on Wireless Communications, Networking and Mobile Computing*. Dalian, China, pp. 1-4.

Lam K. L., et al., 2008. ZigBee Automatic Meter Reading System – Beeline of Metering. *Microwave Conferenc.* Macau, China, pp. 1-4.

Thepphaeng S. and Pirak C., 2011. Design and Implementation of Wireless Sensor Network and Protocol for Smart Energy Meter. *International Conference on Circuits, System and Simulation IPCSIT*. Singapore, vol.7, pp 144-149.

Zhu Y. W. et al., 2006. The Design of Wireless Sensor Network System Based on ZigBee Technology for Greenhouse. *International Symposium on Instrumentation Science and Technology*. Harbin, China , pp. 1195-1199.

Yanfei L., et al., 2009. An improved design of ZigBee Wireless Sensor Network. *2nd IEEE International Conference on Computer Science and Information Technology*. Beijing, China, pp. 515-518.

Shah P., et al., 2008. Power Management Using ZigBee Wireless Sensor Network. *First International Conference on Emerging Trends in Engineering and Technology*. Nagpur, India, pp. 242-245.

ZigBee Alliance, 2002. *Available on: www.zigbee.org*. San Ramon, California, USA.

Froehlich J., et al., 2011. Disaggregated end-use Energy sensing for the smart grid. *IEEE Pervasive Computing magazines,* vol. 10, no. 1, pp. 28-39.

Froehlich J., et al., 2009. HydroSense: Infrastructure-Mediated Single-Point Sensing of Whole-Home Water Activity. *Proc. UbiComp, ACM Press*, pp. 235–244.

Kempton W. and Layne L., 1994. The Consumer's Energy Analysis Environment. *Energy Policy*, v22 n10, pp. 857-866.

Cohn G., et al., 2010. GasSense: Appliance-Level, Single-Point Sensing of Gas Activity in the Home. *Proc. Pervasive, Springer-Verlag*, pp. 265–282.

Haque U., et al., 2011. Pachube. *Available on: https://pachube.com/about_us*. London, UK.

Embit, 2011. *Available on: http://www.embit.eu*. Modena, Italy.

Web Server Comparisons, 1998. *Available on: http://www.acme.com/software/thttpd/benchmarks.html.*