

Smart Grid Applications Using Sensor Web Services

by

Omar Asad

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.Sc. degree in
Computer Science

School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

© Omar Asad, Ottawa, Canada, 2011

To my father soul...

Abstract

Sensor network web services have recently emerged as promising tools to provide remote management, data collection and querying capabilities for sensor networks. They can be utilized in a large number of fields among which Demand-Side Energy Management (DSEM) is an important application area that has become possible with the smart electrical power grid. DSEM applications generally aim to reduce the cost and the amount of power consumption. In the traditional power grid, DSEM has not been implemented widely due to the large number of households and lack of fine-grained automation tools. However by employing intelligent devices and implementing communication infrastructure among these devices, the smart grid will renovate the existing power grid and it will enable a wide variety of DSEM applications. In this thesis, we analyze various DSEM scenarios that become available with sensor network web services. We assume a smart home with a Wireless Sensor Network (WSN) where the sensors are mounted on the appliances and they are able to run web services. The web server retrieves data from the appliances via the web services running on the sensor nodes. These data can be stored in a database after processing, where the database can be accessed by the utility, as well as the inhabitants of the smart home. We show that our implementation is efficient in terms of running time. Moreover, the message sizes and the implementation code is quite small which makes it suitable for the memory-limited sensor nodes. Furthermore, we show the application scenarios introduced in the thesis provide energy saving for the smart home.

Acknowledgements

First of all, I would like to deeply thank my Supervisor Professor Hussein Mouftah, for his unfathomable guidance and continues encouragement throughout the length of my master program study. It was him who gave me this great opportunity to pursue my graduate study at the school of Information Technology and Engineering at University of Ottawa, and explore the promising field of smart grid technology. I learned a lot from Professor Mouftah, he opened my eyes to the beauty of the scientific research, and made me proud of what I did in my thesis; and for that I am incredibly grateful.

I would like to acknowledge Dr Melike Erol-Kantarci's role and assistance along the course of my research; add to that her much appreciated cooperation and suggestions. I had the great fortune of working with Dr Melike from whom I have very much benefited, given her exceptional knowledge on the Smart Grid technology.

Finally, I wish to thank my family who aided me and gave me the hope and love to conclude my studies. In particular my late father who passed away as I was penning this thesis. He has always offered me nothing less than unrelenting support and encouragement throughout my life, and constantly incited my very best. No words could ever express my gratitude to him. All thanks to my mother for her patience and countenance, as well as the adoration of my lovely brothers and sisters.

Contents

1	Introduction	1
1.1	Background	1
1.2	Motivation	4
1.3	Thesis Objective	4
1.4	Thesis Contributions	5
1.5	Thesis Outline	6
2	Previous Work on Sensor Web Services and Smart Grid	7
2.1	Introduction	7
2.2	Sensor Web Services	7
2.3	Smart Grid And Smart Home Technologies	16
2.3.1	Smart Grid concepts	18
2.3.1.1	Advanced Metering Infrastructure	18
2.3.1.2	Demand Response	19
2.3.1.3	Grid Optimization	19
2.3.1.4	Renewable Energy Integration	20
2.3.1.5	Energy Storage	20
2.3.2	Smart Grid Researches Projects	20
2.4	Integrating Sensor Web Services in smart grid	24
2.5	Summary	26

3	Network Architecture and System Analysis	28
3.1	Introduction	28
3.2	System Model	28
3.2.1	Smart Home Components	29
3.2.1.1	Smart Appliance	29
3.2.1.2	Sensor Nodes	30
3.2.1.3	Network Medium	31
3.2.1.4	Web Server	32
3.3	System Analysis	32
3.3.1	Applications Description	33
3.3.1.1	Reading the home meter remotely for billing	33
3.3.1.2	Remote Appliances Control	34
3.3.1.3	Managing Smart Homes Renewable Energy	36
3.4	Summary	38
4	System Design and Implementation	40
4.1	Introduction	40
4.2	Application Implementation	40
4.2.1	Data generator file	42
4.3	J2EE Architecture	43
4.4	Data Base design and Tables	44
4.4.1	Database Tables	44
4.4.1.1	Database Relations	46
4.4.2	JDBC Connector	46
4.5	Interaction Diagrams	46
4.5.1	Sequence Diagram	47
4.5.1.1	Reading Meter Remotely Design Sequence Diagram	47
4.5.1.2	Remote Appliances Control Sequence Diagram	48

4.5.1.3	Managing Smart Homes Renewable Energy Sequence Diagram	50
4.5.2	Activity diagram	51
4.5.2.1	Reading Meter Remotely Design Activity Diagram	51
4.5.2.2	Remote Appliances Control Activity Diagram	52
4.5.2.3	Managing Smart Homes Renewable Energy Activity Diagram	53
4.6	Applications Algorithms	54
4.6.0.4	Reading Meter Remotely Algorithm	55
4.6.1	Remote Appliances Control Algorithm	55
4.6.2	Managing Smart Homes Renewable Energy Algorithm	57
4.6.3	Summary	59
5	Results and System Performance	60
5.1	Introduction	60
5.2	Metrics Measures	61
5.3	Sensor Node code sizes	61
5.3.1	Reading Meter Remotely Messages	62
5.3.2	Remote Appliances Control Messages	63
5.3.3	Managing Renewable Energy Completion Time	65
5.4	Sensor Node WSDL Files	66
5.4.1	WSDL for Reading Meter Remotely	66
5.4.2	WSDL for Remote Appliances Control	68
5.4.3	WSDL for Managing Renewable Energy	73
5.5	Completion Time	75
5.6	Energy Saving	77
5.6.1	Remote Appliances Control Energy Saving	78
5.6.2	Managing Renewable Energy Savings	80

5.7 Summary	82
6 Conclusions	84
6.1 Summary and Concluding Remarks	84
6.2 Future Research	86

List of Tables

Table 5.1: The code & messages sizes(Bytes)	75
Table 5.2: Summary of the Completion values distribution	77

List of Figures

Figure 2.1: TCP Message exchange latencies[1]

Figure 2.2: TCP Message exchange latencies with delayed acknowledgments disabled[1]

Figure 2.3: TCP Message exchange latencies with delayed acknowledgments disabled[1]

Figure 2.4: Extended Web Service Technology Stack[2]

Figure 2.5: Smart Grid Concepts[3]

Figure 2.6: Industry drivers of smart grid[4]

Figure 2.7: Technical measures impact energy efficiency and efficient network management[5]

Figure 3.1: Smart grid applications model using sensor network web services

Figure 3.2: Smart home sensor wireless network

Figure 3.3: Constant vs. Fluctuating Electricity Rates in Ontario [6]

Figure 3.4: A logistic distribution shaped for Oil production curve [7]

Figure 4.1: Java NetBeans Framework for Sensor Web Services Applications

Figure 4.2: Java NetBeans Framework for Sensor Web Services Applications [8]

Figure 4.3: Mysql Data base tables used for the applications

Figure 4.4: JDBC connector to MySql from Java

Figure 4.5: Sequence diagram showing the process of Reading Meter Remotely

Figure 4.6: Sequence diagram showing the process of Remote Appliances Control

Figure 4.7: Sequence diagram showing the process of Managing Smart Homes Renewable Energy

Figure 4.8: Activity diagram showing the process of Reading Meter Remotely

Figure 4.9: Sequence diagram showing the process of Remote Appliances Control Activity

Figure 4.10: Activity diagram showing the process of Managing Smart Homes Renewable Energy

Figure 5.1: Completion time for A single process in Sensor WS application

Figure 5.2: Completion time for the Reading Meter Remotely application

Figure 5.3: Completion time for the Remote Appliances Control application

Figure 5.4: Completion time for Managing Renewable Energy application

Figure 5.5: The amount of energy saving by running the Remote Appliances Control application.

Figure 5.6: The amount of energy generating from the Renewable resources results from running Managing Smart Homes Renewable Energy

Abbreviations

WSN	Wirless Sensor Network
WS	Web Services
WSDL	Web Services Description Language
SOAP	Simple Object Access Protocol
REST	Representational State Transfer
TCP/IP	Transmission Control Protocol/Internet Protocol
WWW	World Wide Web
W3C	World Wide Web Consortium
PV	Photovoltaic
JDBC	Java Database Connectivity
PC	Personal Computer
SCM	SOAP Compressed Message
LTP	Lean Transfer Protocol
BPEL	Business Process Execution Language
DINS	Data Intense Network
AC	Air Condition
MDM	Meter Data Management
CIS	Customer Care and Billing Systems
OMS	Outage Management Systems
HTTP	Hypertext Transfer Protocol
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
KWH	kilowatt hour
J2EE	Java 2 Enterprize Edition
PHEV	Plug-in Hybrid Electrical Vehicles

Chapter 1

Introduction

1.1 Background

Despite the wide spread of technology on different aspects of life, the electricity grid does not benefit from the revolution in the IT sector. Moreover, the current grid delivers the electricity in the traditional way, where the home is considered as a black box that merely absorbs electricity, and that in no way manages the usage of home appliances. This results in energy lost, and increases the load on the utility generators.

The current grid experiences many drawbacks and has many shortages, which in turn affects the services provided to end users:

- Lack of information available on the utility system on how much electricity the home consumes at a certain time during the day.
- No way to communicate with the home appliances and control its usage.
- Demand increasing in high percentage on the peak hours .
- No systems to control the integration of renewable energy resources with the current energy from the grid.

- The emergence of the plug-in hybrid electric vehicle (PHEV) which needs to be charged.

The need to renovate the current energy grid so as to maintain its capabilities to deliver electricity, and to accommodate with the emerging techniques is vital. The integration of advanced technologies and telecommunication infrastructures on the power grid is called the Smart Grid.

The smart grid aims to save energy, reduce cost and increase reliability and transparency on the consumer side. On the other hand, it helps utilities manage their electricity generators and keep track of consumer consumption. Furthermore, the smart grid aims to reform the relation between the consumer and energy, by changing consumer habits in order to be an active element on the process of generation and distribution. For example, the consumers would have the ability to generate their house energy needs locally through renewable resources like solar panel or wind turbine, and could also sell the extra electricity back to the grid.

From the technology perspective, there is a need to add a real intelligence to the existing power grid, so that the utility could control the delivery of the power over the grid, and take a real time action when needed. As a result, the advancement of the smart grid revolution, realizes the establishment of a communication infrastructure, in addition to the deployment of the sensor technology and IT systems which help to optimize the performance of the grid and improving security and reliability[9].

In order to add information propagating flexibility to the grid, Wireless Sensor Networks (WSN) are used. The sensor nodes measure and sense data from the end points on the grid and send it to utility machines to take proper action.

WSN, consists of distributed sensor nodes, communicating with each other using a communication protocol like ZigBee. Each sensor is able to sense an environmental or physical phenomenon, such as temperature or humidity. Mainly, each node on the WSN perform two tasks, sensing the data as well as propagating the messages to the other nodes.

In order to exchange the data between sensor nodes and between a sensor node and an external machine, sensor Web Services (WS) is an efficient tool to rely on. A server that is run by the utility, and that could access the sensor web service, asked to get specific information, and send an information for the sensor to take an action.

WS is a software that supports machine to machine interaction over a communication network. The web service publishes itself as an interface called a WS Description Language (WSDL), the client accesses the web service and exchanges messages with it through Simple Object Access Protocol (SOAP), the SOAP message uses the XML structure.

The deployments of WS in the resource constrains devices like a sensor node has some difficulties, first the sensor processor capabilities and the memory size are quiet small to handle the WS messages and required data. In addition, the WS SOAP messages and the XML file size can cause significant overhead to the sensor and does not fit its capabilities.

Some techniques and solutions have been investigated to customize the web service and its components in order to fit the limitation of the sensor nodes. In addition, several algorithms solved the compression in XML file to produce a smaller file size. Furthermore, many techniques used to reduce the code size on the sensor side, and make it suitable to the sensor node. In addition, the network and application layers have been redesigned, by optimizing the TCP/IP protocol and minimizing layer messages sizes.

On the other hand, using WS in sensor node has advantages. First, it is machine independent, which allows different platform running WS to communicate with each other. Second, the data format used by web service is easy to understand and to deal with. Third, they are versatile enough to reuse in different systems.

1.2 Motivation

One of the important aspects of the smart grid is the novel applications. The applications have the ability to control the flow of power in the grid, manage the smart home appliances usage, in addition to integrate utility systems and provide the inhabitants the privilege to access those systems to track their house power consumption.

As the smart grid is moving towards reality, many energy technologies started taking place. Even nowadays, there is a lack in the systems that manage those technologies and arrange their integration with the current grid. On the other hand, the use of web technologies to enhance the grid information communication, and to let the inhabitants have easier access to their home information on the utility system, as well as access to their house remotely become necessary.

Nowadays, there is a real need to develop a real time application to read the current meter reading remotely, without wasting human time and effort, and having more accurate readings. Furthermore, the high demand during the peak hours, will affect the utility generators to provide the energy to the homes, therefore, the development of an application to monitor the homes appliances on the peak hours becomes vital. Moreover, while renewable energy equipment are being installed in the houses, there are no system to arrange the usage of the power from the renewable resources or from the utility.

1.3 Thesis Objective

The main objective of this thesis is to develop systems for three smart grid applications using sensor WS, and measure their performance and energy efficiency. Application are:

- Reading Smart Meter Remotely: by allowing the utility server to get the current homes meter readings, and their time stamp.
- Remote Appliances Control: At the peak hours, the high energy demand, causes a problem on the utility side, which affects its ability to provide a proper energy to

the customer. Therefore, this application could manage the consumer appliances on the peak hours and decrease the load on the generators.

- **Managing Smart Homes Renewable Energy:** The renewable resources can generate energy to the residential homes, the users could use the renewable resources electricity, the utility electricity, or they may sell the extra saved energy back to the grid. This system manages the use of the renewable resources, and to select which one of the mentioned choice to take.

1.4 Thesis Contributions

The contributions of this thesis are the following:

- Develop three smart grid applications to reduce energy consumption and to enhance utility information management. The Following are the applications which have been developed:
 - Reading the home meter remotely: By letting the utility web server access the inhabitant's home and read the current meter reading, and save it on the database with the current time stamp, to enable the user as well as the utility to browse those readings.
 - Remote Appliances Control: This application enables the utility system to control the inhabitant's appliances during peak hours, through getting the current appliance electricity consumption, and depending on a certain algorithm, it determines whether the appliance has to decrease its consumption.
 - Managing Smart Homes Renewable Energy: the current application manages the usage of the renewable energy in the home. Once the smart home depends on the renewable resources in addition to the utility energy. The application will then determine when to use the stored energy from the renewable, or

switch to use energy from the utility, or sell the extra stored energy back to the grid.

- Integrate Sensor WS technology into smart grid applications. By deploying a sensor network in the smart home, each sensor node runs a web service that performs a specific task, the first sensor is combined with the home meter and is capable to request the current meter reading and sends the result through another sensor node acting like a gateway to the utility web server. In addition, another sensor is combined to the appliance runs a WS to read the consumption and sends it to the web server which may send a request to another WS running on the same sensor node to decrease the appliance consumption. The last WS is running in a sensor node combined with the storage device to detect the amount of energy available on the device and take the proper action.

1.5 Thesis Outline

This thesis consists of six chapters. Chapter 1 introduces the background knowledge in the area of the smart grid applications and sensor WS. In addition, this chapter states the motivation, the objective, and the thesis contribution. Chapter 2 gives an overview about the technologies and concepts related to the research, presenting sensor WS implementation and limitations. In addition, several smart grid systems and applications are discussed. Chapter 3 presents the network architecture and system analysis. The chapter includes the system model as well as the application descriptions. Chapter 4 describes the system designed and implementation. The chapter also presents the algorithms used to develop the applications. Chapter 5, presents the results and performance measurement, mainly the energy saving and system performance. Chapter 6 presents the conclusion of the thesis and future research.

Chapter 2

Previous Work on Sensor Web Services and Smart Grid

2.1 Introduction

This chapter gives an overview about the technologies and concepts related to this research. The next sections present the works that have been done on the sensor WS network and smart grid applications. In Section 2.2, an overview of the sensor web services implementation and limitations is presented. Smart grid applications and systems are discussed in section 2.3. Section 2.4 presents the integration of the sensor WS network in the smart grid, showing the ability of implementing WS in sensor nodes. Lastly, section 2.5 gives a summary.

2.2 Sensor Web Services

The World Wide Web Consortium(W3C)[10], defined WS as: “a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with a WS in a manner prescribed by its description using SOAP messages,

typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.”

WS has emerged as a widely used tool in the web enterprise application. One of the major benefits of the WS is that the platform is independent, which makes it ideal for large system environment like the Internet. Furthermore, the protocols and data format are easy for developers to understand and deal with. And it is flexible enough to allow the reuse of services in different systems.

Because of the above reasons, WS seem like a promising tool to be used on the resource constraint devices like a sensor node, since they have the ability to provide remote control, data collection, and queering capabilities. In addition to that, the integration of the WS in sensor nodes have the following advantages [1]:

- 1 : The interoperability with other applications. The applications can be done in a flexible manner allowing integration with other applications easily. Therefore the developer does not have to know about the application implementation, the only thing he/she needs to know is the interface of the application, and the parameters used.
- 2 : Improving the development of the system. The WSDL file generated by the WS, has all the information related to the WS implementation, such as the sensor address, the method name and parameters. Hence, it can be easily used by a high level language development tool like NetBeans IDE and .Net.
- 3 : Ease of integration with enterprise systems. Since many Internet applications have been built using WS. It is easy to use the sensor node WS to integrate with any web applications. In that way, many physical sensing information will be available online for the Internet users. The idea of the Sensor web becomes available. In our application, the home owner have the ability to access his/her home from outside to query information about the electricity consumption.

5 : Implementing the WS on the sensor node directly, eliminates the use of the gateway to communicate with the sensor node, because all the messages and protocol have the same format on both the sensor side and machine side, and do not need to convert between the device platform.

On the other hand, using WS on resource-constrained devices, such as sensor nodes, may experience several challenges. First, the payload of the WS messages consists of XML data, TCP/IP, and HTTP. They easily exceed the sensor node capabilities. Secondly, even XML is a wide support messaging tool and its even human readable, it needs a larger memory size comparable with small sensor node size; (for example, The Pace-mate sensor network hardware platform [2] has a 32 KB RAM). Furthermore the size of the code and compiler have to fit in the sensor memory.

In the following, recent research studies have been done to address the above challenges, and to make the WS implementable in sensor nodes. The main goal of these studies, is to customize the WS to fit the sensor node limitations.

In order to customize the WS, and make it suitable for resources constraint devices, we need to rebuild the components and protocols composing the WS. The major components are: XML data structure, Web Service Description Language (WSDL)[11] , and TCP/IP protocol.

In the following, we will present the related works that figured out the integration of the sensor in the world wide web, the compression of the XML file, eliminating the unnecessary part of the WSDL file and reformat it, and the interoperability of the network layer in sensor nodes using TCP/IP.

IrisNet [12], presented an architecture to connect a heterogenous sensor network to the web, therefore the user can access, from an internet connected device, millions of distributed nodes all over the world and get specific data. Their work was implemented depending on a gateway node, which is a non-constraint device. This model was developed depending on the sensor web idea[13, 14]. The sensor web simply, is a set of distributed, wirelessly connected sensor nodes, that can be deployed to monitor and ex-

plore the environment. However, the term sensor node refers sometimes to the sensor node connected to the World Wide Web (WWW). The system implemented in our work is a sensor web. Since the nodes residing in the home, they have the ability to collect the data from appliances, and ask the appliance to carry out specific tasks. Furthermore, all the sensors are connected through the internet with a web server, which can be accessed by the user.

XML is the data type composed the WS messages[10]. It is a widely used, human readable, and a machine independent language, which makes it an ideal tool for multiple platforms like the internet. On the other hand, using XML on the sensor node still has problems, because of the big file size the xml generates, where the sensor node does not have the ability to process it. Therefore, some studies have been done to compress the XML file, and minimize its size and payload; Xenia[15], is an xml compression tool used to reduce the size of the xml file depending on the xml schema description at compilation time. Another work was done to compress the XML file in the Wireless Sensor Network (WSN) is microFibre, presented in[16], this tool does not compress the XML scheme only, but it also compresses the XML bit representation in the memory. In addition to that, microFibre is platform independent, which means it can be used in gateways, as well as in sensor nodes.

SensorML[17], gives a description of a sensor node system in xml language. The processes described by the xml can be discovered and executed, also all the process define their input, methods, and outputs. Other works have been to come up with new tools to compress XML file, which can be used to compress the xml file in the sensor node. Augeri et al[18], gives an analysis for the XML compressor tools, and identifying the key factors when choosing the compressor. XML tree structure compression was discussed in[19], the idea of their work was to compress the tree structure part of xml document, not the data content. The authors used a known algorithm to derive the grammar of the tree structure, depending on the repetition of the tree pattern. However, in our implementation we used the XML optimization techniques proposed in[1]. This method

replaces some of the method names and argument names which are defined by the sensor node, with very compact names. For instance, the method name (`getMeterReading`), which is 15 bytes can be replaced with a another smaller method name like (`GtMetRdng`), which is 9 bytes. Further details will be discussed in the results and performance chapter.

As we mentioned, the implementation of WS in the sensor node has to redesign the network layer protocols, therefore both the sensor node and the connected device can communicate with each other and transmit messages. Eight bit microcontroller TCP/IP protocol was presented in[20]. The new protocol keeps the same functionality as the previous one, but it minimizes the interface between the TCP/IP stack and the application. In[21], the authors showed how to customize the TCP/IP protocol to fit the wireless sensor networks. They introduced the spatial IP address assignment scheme for the WSN, by providing a semi-unique IP address to the sensor nodes. However, we are working on the application layer, and for the network protocol we assume that we used the 8 bit TCP/IP protocol presented in[20], and the IP address for the sensor node is manually configured.

Priyantha[1], presented a WS design approach to enable the development of the sensor systems. The approach design was flexible, to allow additional nodes to add to the initial deployment. The authors tried to redesign the network and the application layers to suite the sensor networks. The overhead of the TCP/IP which includes, the number of bytes sent, the number of packets sent, and the latency due to TCP/IP, is suppose extremely large (to send a 10 B of raw data with TCP/IP, the packet has to carry 40 B of TCP/IP header). Therefore, the authors have come up with some techniques to optimize the TCP/IP protocol to fit the sensor node:

- 1 :Persistent TCP connections. Since small number of the application will be using the sensor node WS, it is more efficient for the client application to maintain an open connection with the sensor node. So, there is no need to establish a new connection every time the client requests data from the sensor WS. Hence, the number of messages needed for the TCP/IP transfer will decrease. In Figure 2.1, The ac-

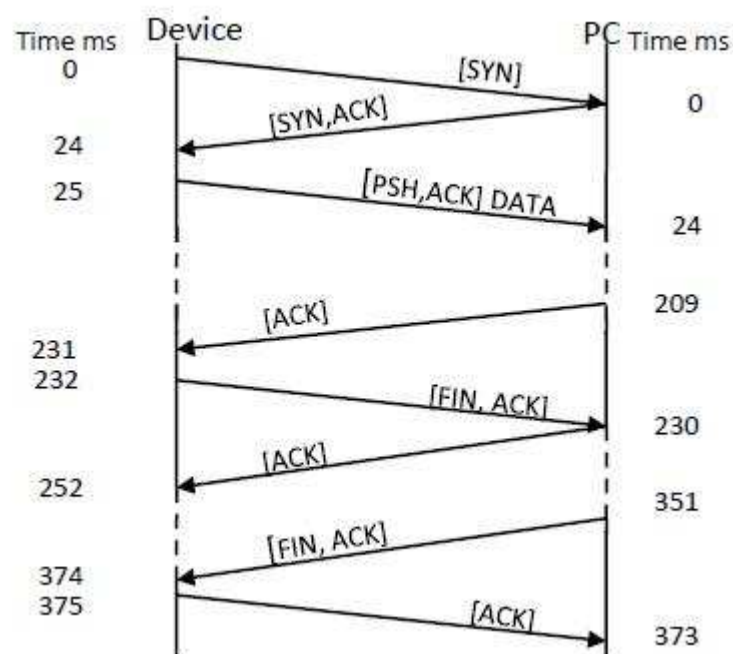


Figure 2.1: TCP Message exchange latencies[1]

knowledge message (ACK), is not needed in the persistent TCP, therefore, the latency reduced by $(231-25)=206$ ms.

2 :Disable delayed TCP acknowledgment: In the regular TCP, for every packet received at host, the TCP starts 200 ms timer before sending the acknowledgment, if another packet received before the time expires, the acknowledgment will be sent. Otherwise, the acknowledgment will be sent after the timer expires. However, this mechanism works fine for the congested network, but in sensor network WS, just few messages are sent over the network, so, disabling TCP acknowledgment message delay will reduce the TCP latency. Figure 2.2 shows the same message exchange in Figure 2.1 after disabling the acknowledgment delay.

3 :Link layer retransmission: The TCP protocol uses a retransmission mechanism to achieve end-to-end reliability, but this can cause a significant delay. In Figure 2.3, the packet number 5 is lost, so the TCP retransmit it in $(2989-20)=2969$ ms. In

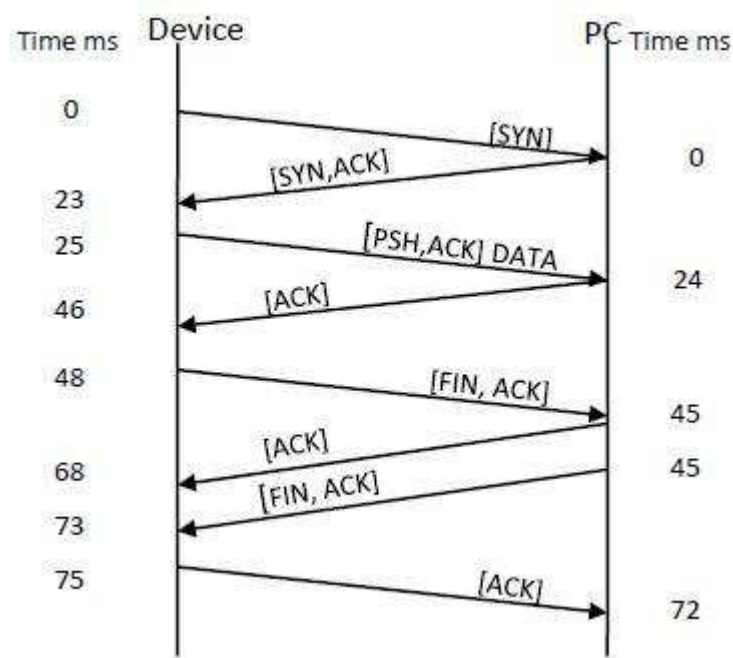


Figure 2.2: TCP Message exchange latencies with delayed acknowledgments disabled[1]

wireless networks, it is not efficient to use this mechanism, since the probability of losing packets is higher than the wired network. Therefore we can use link-layer Automatic Repeat Request (ARQ), such as[22]: Stop and Wait, Go back n, and Selective repeat. Using those techniques, will ensure faster connection, and avoiding the TCP timeout.

4 : Low power mode between TCP messages: Keeping the sensor radio signals on between packets sending, will cause an energy overhead. To reduce the energy loss, the radio is turning off between TCP packets transmission.

5 : Link layer fragmentation: This can be done by segmenting each packet into smaller packets and send it to reduce message delivery times and power consumption.

The integration of the sensor node into a business process has been presented in[2].

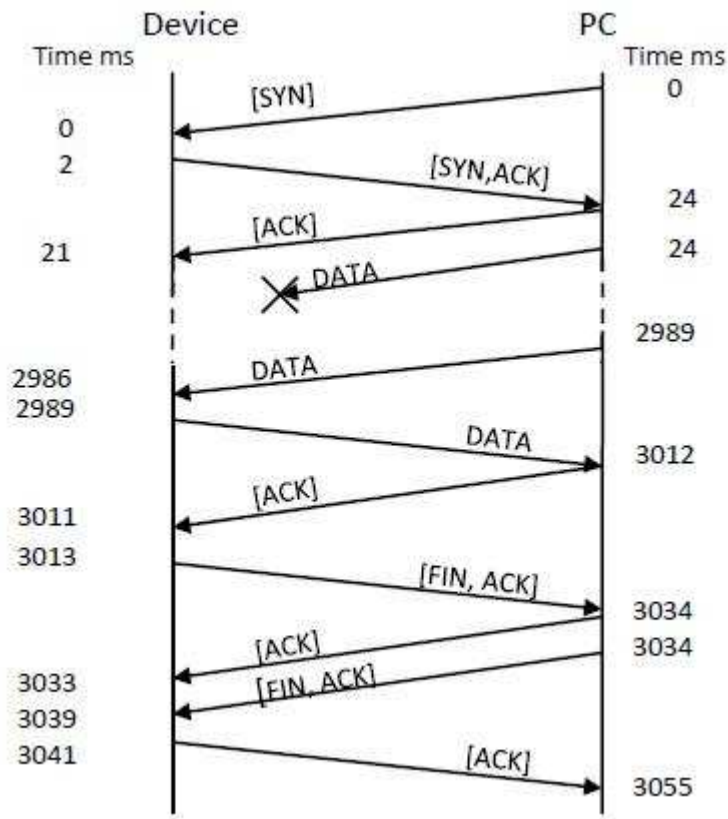


Figure 2.3: TCP Message exchange latencies with delayed acknowledgments disabled[1]

Business process is defined as a set of activities, where each activity is implemented using a WS. The authors have used the Business Process Execution Language (BPEL)[23] language for a sensor node that is performing a business process. In addition to that, they have introduced a SOAP-compressed message(SCM) technique, and a Lean Transfer Protocol(LTP). Figure2.4, presents the WS standard stack, and the SCM and LTP are highlighted to show how it is fit in the original stack. In this research, our system is performing business process, as well, since each application is performing a task, like reading meter or remote appliance control. However, instead of using PBEL we use a NetBeans IDE to implement the WS, since its an efficient and easy tool to deal with WS.

Two protocols could be used in the WS are: Simple Object Access Protocol SOAP

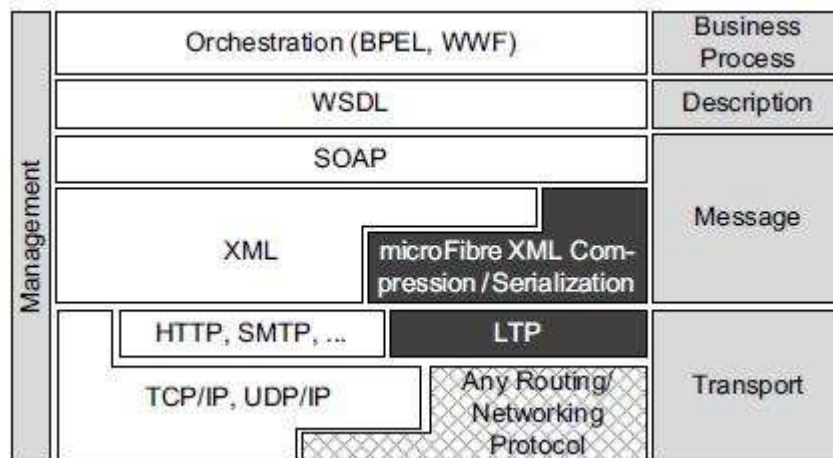


Figure 2.4: Extended Web Service Technology Stack[2]

and Representational State Transfer REST. SOAP[24] is a lightweight protocol used to exchange structured information in a decentralized environment. It uses XML file format to exchange the data. On the other hand, REST[25], is also a lightweight protocol used to exchange the data in a decentralized environment. However, REST does not use XML as a data format, and it is a stateless protocol. Using RESTful WS on an IP-multi hop network, was investigated in[26]. Furthermore, the authors compared the efficiency and power consumption of using RESTful versus SOAP, showing that using RESTful is more efficient than SOAP. However, SOAP is still a widely used protocol in WS and compatible with other applications, furthermore it is easy to implement and use, since it is the default WS protocol for many IDE.

Other studies have been done to come up with a service oriented programming model for sensor network. Amundson et al [27], presented a model which can add and access any WS. They assume the sensor network application as a well-defined interfaces graph of modular and autonomous service, that allow them to be described, published, discovered, and invoked over the network. This will provide a convenient way for integrating services from heterogeneous sensor systems.

In[28], the authors presented a communication mechanism architecture on the low

power devices. They depend on the RESTfull based on HTTP to implement a WS. In addition, they presented an architecture of emerging Bluetooth Low energy technology with RESTfull device for transport connectivity.

To solve the problem of multi-user data queries in Internet-based Data Intensive Sensor Network DINS, Takahashi and Tang [29] developed a LiteWS, a WS system that aims to enhance multi user queries in DINS. The idea of LiteWS based on using a simple caching technique to handle multi requests, and to reduce packet loss ratio in the system. In our work, we design the system in order to let the server access one node at the same time, so no congestion will occur.

2.3 Smart Grid And Smart Home Technologies

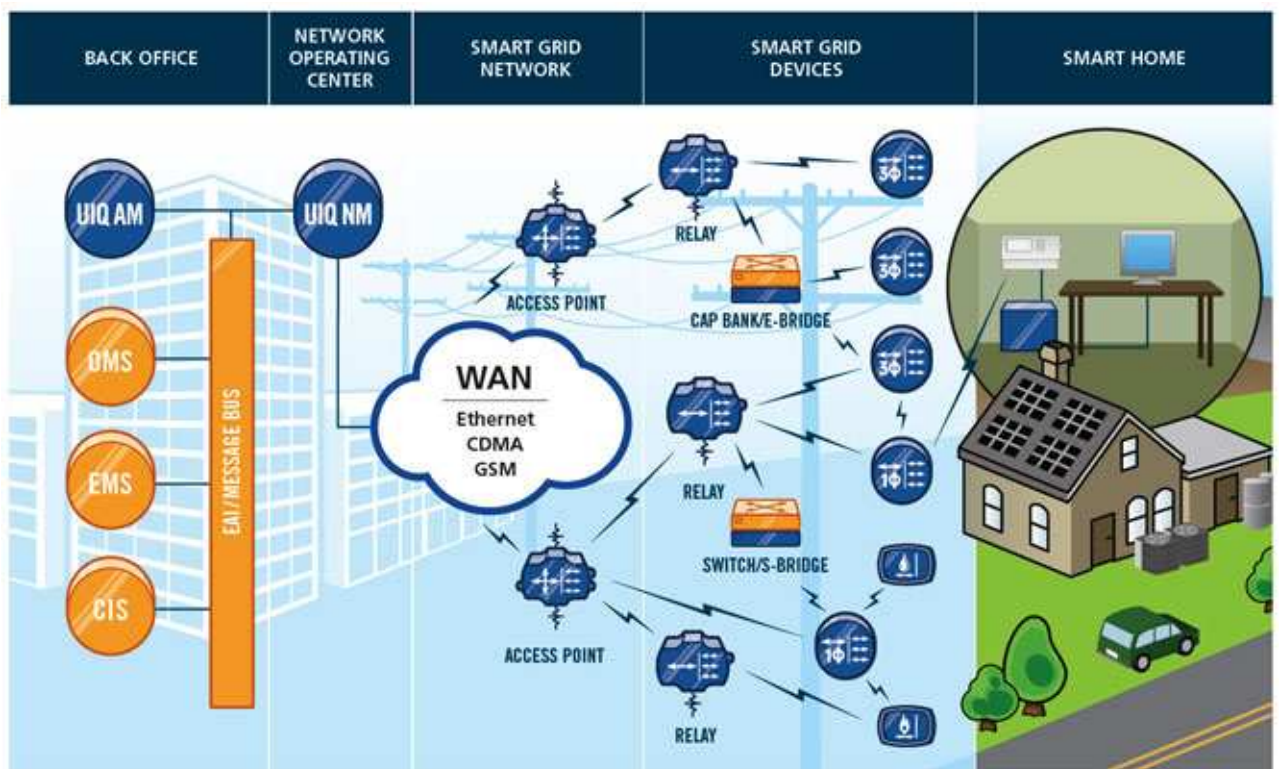


Figure 2.5: Smart Grid Concepts[3]

In the last few decades, the current power grid faces several challenges. First, the demand for energy is increasing. It is expected that the amount of energy demand will double the current amount by 2020[30]. Second, fossil fuels resources are decreasing, and the world is seeking other energy resources. Third, the amount of CO₂ produced by burning the fossil fuels is becoming very high and must decrease.

Therefore, the need to upgrade and enhance the current energy grid to face the above challenges becomes vital. A Smart Grid[3] has the ability to generate and distribute electricity more effectively, economically, securely, and sustainably. The smart grid integrates the current technology and tools into the power grid starting from generation, transmission and distribution all the way to the customer homes, which contains a smart appliances and advanced sensing tools. Furthermore, a two way communication mechanism involved to control and monitor the home electricity behavior. Figure 2.5, illustrates the main components of the smart grid and its interaction.

The implementation of the smart grid contains three main steps which can be summarized as:

- 1 : installation of smart meters that has two-way communications capability with the utility and the consumers.
- 2 : implementing network infrastructure to connect all the participating units and deliver data in real-time.
- 3 : developing applications that can manage the grid components.

Among these steps, installation of smart meters has started in North America, Europe and China, and its planned to be due by 2012. Smart meters can collect, measure, process, and analyze the energy usage of a consumer, they allow remote meter reading and they can wirelessly communicate with the other grid components [9]. On the other hand, integration of sensor networks with the smart grid provides new applications and services. For instance, appliances can be accessed remotely by the inhabitants and their power consumption can be controlled. Moreover, utilities may communicate with the

sensors and query some necessary information affecting the grid operation, when the consumers have opt in for programs allowing utility's access to their appliances. Sensor network WS are promising tools for these new Demand Side Energy Management (DSEM) applications.

2.3.1 Smart Grid concepts

Smart grid has started to be an attractive topic for both the researchers, industries, and the governments. This section will discuss concepts of Smart Grid applications and technologies. The rest of this section, we will give an overview about some technologies and studies have been emerging in smart grid area, focusing on the system and schemas developed to act a key role in the smart grid technology.

2.3.1.1 Advanced Metering Infrastructure

Advanced Metering Infrastructure (AMI)[9], defined as the system that can deal with the energy usage, by sending the data over a two-way communication network, connecting the smart meter with the utility control systems. Therefore, the AMI has two components:

- 1 :The physical smart meter.
- 2 : The communication network that can transport the data from the smart home to the utility control system and vice versa.

AMI has two layers: The application layer, and the transportation layer. The application layer is responsible of collecting data, monitoring, controlling the entire grid through the systems, which has the ability to receive data from numerous homes and analyze it, to take appropriate action to monitor and control the grid, in addition to deliver an efficient power, to achieve power utilization, and to ensure reliability and security. In our applications , we are focusing on the application layer, by developing applications, that can effectively integrate in both the smart homes and the utility, in order to achieve energy saving, as well as increase user comfort.

On the other hand, transport layer is concerned with moving the data from the utility to the end user and vice versa. However, we suppose that communication network is already installed and the data can be transport from one system to another.

2.3.1.2 Demand Response

Demand response aims to reduce the end users energy consumption during the high demand time, that happens in some periods of the day called the peak hours. With the advancements of the Smart grid technologies, the residential users will have the opportunity to participate in a demand response program, depending on a contract made in advance which determines when and how to reduce the user load, the utility systems will have the ability to manage the end user appliances and adjust the smart user devices, therefore, the utility generators keep providing its desired service in the peak hours.

Since the Demand response is a key factor in the smart grid technologies, we developed an application to manage the smart home Air Conditioning on the peak hours.

2.3.1.3 Grid Optimization

Grid optimization can be employed by installing a wide instruments through the grid, which gives the utility digital control over the power grid. Improving the efficiency, security, and reliability, can be done by adding sensor technology, communication infrastructure, in addition to the information technology systems. The deployment of large scale sensor networks in the power grid is valuable and important, since it has the ability to collect a large amount of data from all over the grid and send it to the utility IT system. On the other hand, the IT system in the utility gathered the sensing data across the grid and converts it to actionable information. Our smart grid applications deployed in a sensor network environment, assuming that the smart home has sensor nodes combined with the smart appliances, and the utility or the home owner could interact with those nodes through a web server.

2.3.1.4 Renewable Energy Integration

One of the smart grid aspects is the integration of the renewable energy and the distributed generation sources, since using the natural resources, as wind and solar, to produce the electricity could save a significant amount of the energy produced from fuel. In such a situation, the future smart home would have one or more of renewable resources to cover the home electricity needs, and to sell the extra amount back to the grid. Although, renewable energy has been used for long time, the management systems and the grid infrastructure that enable the renewable to integrate in future smart grid have not introduced yet. In our work, we developed Managing Smart Homes Renewable Energy system, which can control the house energy usage getting from either the grid or the locally generated energy.

2.3.1.5 Energy Storage

The smart grid revolution aims to reduce the power consumption and benefit from the renewable energy. Since the renewable resources are variable from time to time, depending on the weather situation and the geographical location, the energy produced when there is a little demand should be saved. The need to have an energy storage does not come from the fact of storing energy only[9], but it helps as a backup energy source to the grid, and serve as an alternative to fossil based generation. In our system we assume the existence of an energy storage device to store the renewable produced energy.

2.3.2 Smart Grid Researches Projects

In the smart grid, the home owner can participate effectively in his/her house energy behavior. Many projects[31, 32], are working now to enable the end users generating a clean environment energy. The NOW house project[31], is a leading project that aims to retrofit the current old houses in order to produce their annual energy needs. The idea of the project is the following. The NOW house still connected with the grid to send

the energy that is generated, by the end of the year, the amount of the energy sent back to the grid is equal to the energy the house consumed from the utility, resulting a zero energy bill.

On the other hand, the Feed-In-Tariff is a policy designed by the utilities to encourage people to use the renewable energy. Ontario government Feed in Tariff[32], urges the people to develop renewable energy projects across Ontario, and sell the generated electricity back to the grid with a fixed price in 20 years contract. Some researches have



Figure 2.6: Industry drivers of smart grid[4]

been done to manage the home appliances energy consumption. Smart-a project [33], is a European project focusing on demand side load management. By identifying and evaluating the potential demand that arises from coordinating energy demand of utility appliance with local sustainable energy production, by taking in consideration, the requirement of regional load managements in the domestic network.

Demand delay has been discussed in [34], they give a description about the appropriate time to turn on the appliance, such as a washing machine which has three priority stages the user can choose: immediate, complete wash in 4 hours, or complete wash in 12 hours. In [35], the authors design a bottom-up model for demand side management by applying two strategies: either shift the appliance usage to a later time or turn off the appliance.

In [36], an appliance coordination scheme ACORD-FI, is proposed to help the consumers decrease their energy expenses by selecting the appropriate time to run their appliances. The consumer turn on the appliance at any time, that will generate a request packet to the EMU. The waiting time is calculated and sent back to the user, then the user have the choice to wait for the time assigned or turn on the appliance right away. However, [36] does not support remote control and WS applications.

The smart grid paradigm has been discussed in [4], the authors investigate the main industry drivers of smart grid. Figure 2.6 illustrates those drivers, which includes: focusing on environmental protection by using renewable generation, developing better asset utilization, and the need for end user involvement. Furthermore, they discussed the demand response and the market product that can participate in.

In [37], the authors discuss the information infrastructure requirements to store all the phaser measurements from all over the grid, concluding that the current centralized way is inefficient to store all that amount of data. Then, a new application was suggested. The new application classified into two categories: the first is the wide area control, which aims to improve the overall stability of the power system. The second, is the set of controller system to control EMS. However, our work is focusing on developing a management applications for the energy flow, and how to efficiently use the energy resources.

The International Energy Agency [38] arguing, that although Distributed-Generation technology represents a small portion of the electricity market, it still has the ability to be an active source of emergency capacity, and to provide a reliable alternative service. The study also introduced three stages to emerge the distributed power system into the

current grid:

- 1 : Accommodation of the distributed power systems into the current grid.
- 2 : Create a decentralized network system that is compatible with the centralized one.
- 3 : Introduce a new system, most of its power generated by a distributed generation, and a little amount generated by a centralized station.

In our work, we introduce a management system, that can accommodate the house usage of the electricity, whenever it is from the power grid or from the renewable resources(Generation distribution).

Molderink et. al[39], introduce a management plan for domestic power grid, and for the residential houses. After the system model was proposed, they showed the three steps plan:

- 1 : Local Predication: a production system is located in each house is responsible to generate a daily file. The file contains the generation and consumption pattern for each appliance for the coming day, the prediction can be generated using a neural networks, then the local system send this file to a global controller.
- 2 : A central planner receives the energy profile, and determines the potential for each unit to achieve the global objective, noting that the model is based on the tree structure, which means that the house sends its profile up to the root. After this step each house will receive its plane for the upcoming day.
- 3 : The local controller receives the steering plan from the parent, and a real time algorithm scheduled the time the appliances can be turned on/off, and how much energy does it need or consumes.

This application is similar to what we are doing in the controlling appliance algorithm. In our application, a central system is reading the current Air condition (AC) consumption, and sends back whether it needs to decrease its consumption or not depending on the overall consumptions of the houses. However, [39] does not use a sensor network in their algorithm, and they do not support remote control neither WS applications.

2.4 Integrating Sensor Web Services in smart grid

Sensor WS has not been used previously in the smart grid technology, however, implementing WS in a smart grid have been investigated in few works. As the smart home is considered a key part of the future grid, the house appliances need to be organized and communicated with each other and with a central controller device to manage its working time. Sensor Network seems an efficient way to communicate all the home appliances. Furthermore, it can sense the wanted data from the appliance combined to or from the existing environment and send it. On the other hand, WS have the ability to request the desired information and propagate it to the requester (Web server).

In this section we will give an overview about the research which discussed the usage of the WS in the smart grid.

Warner et al [5] have presented the requirements for integrating a WS into a smart grid from technical and management perspective. Figure 2.7, illustrates the technical measures that may appear in the smart grid, and seven impact categories will affect on the future smart grid. Three requirements have to be translated into a business application, in order to affect both energy and network management efficiency:

- 1 : The end user feed back. So the users can keep track of their house behavior and monitors their consumption.
- 2 : Automated decentralized control of distributed and demand response. Like optimized device scheduling depending on the electricity prices and peak hours.

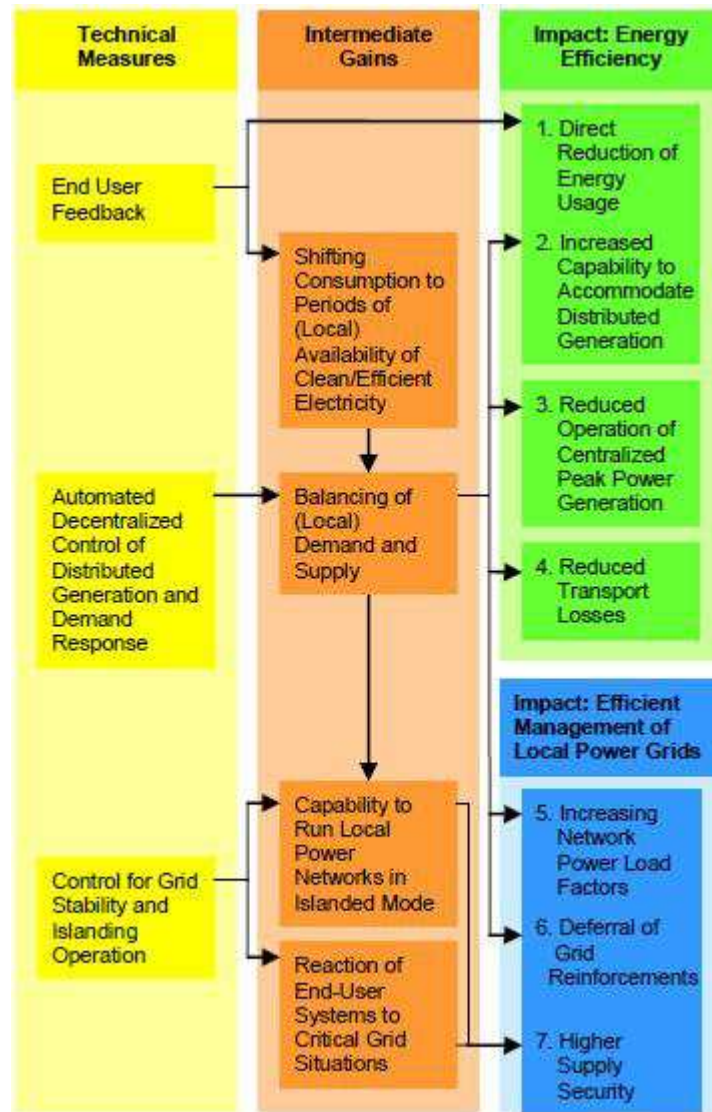


Figure 2.7: Technical measures impact energy efficiency and efficient network management[5]

3 : Control of grid stability and islanding operation. It is an automated operation, aims to deliver the service in critical situation.

In our application, the inhabitant will know his/her house’s energy consumption behavior by accessing the information in the database server.

2.5 Summary

In this chapter, we introduced an overview of sensor WS limitation and implementation details. Then we talked about smart grid applications and technologies. Finally we discussed recent works on using WS in smart grid technologies.

Implementing WS in the sensor node has many advantages. First, WS is platform independent. Second, the protocols and data format are easy for the developers to understand and deal with. Third, its flexible enough to allow the reuse of the service in different systems. On the other hand, implementing WS in the sensor nodes has many challenges due to the resource constraints of the node. To solve this problem, many studies have been done to customize the WS and its components. We presented several techniques to compress the XML file to suit the sensor node. Furthermore, we showed compression techniques to reduce the code and message on the sensor node side. Then we presented an effort to redesign the network and application layers, by optimizing the TCP/IP protocol and minimizing its messages sizes.

The smart grid aims to reduce the electricity consumption, and to increase the reliability and transparency. The idea of the smart grid involved integrating the recent advanced technologies with the current grid to have a two way communication between the utility and the end user home. On the other hand, the end user will be more active on the energy cycle. Furthermore, the smart home has the opportunity to produce clean energy from the renewable resources like wind or solar, to decrease the demand load on the grid and to increase the energy saving. We showed many applications and studies in the smart grid, discussing some systems and schemas developed to manage the electricity consumption

Including WS technology in the smart grid have not been investigated widely in the literature. However, WS seems to be a promising tool to propagate the data between the smart home and its appliances and the utility and vice versa. We discussed the requirements for integrating a WS into a smart grid from a technical and management

perspective.

Chapter 3

Network Architecture and System Analysis

3.1 Introduction

In this Chapter, we will present the Network Architecture and System Analysis. The next section provides details about the system model and the concepts of the environment needed to implement the application. In Section 3.3, the system description and the software analysis are discussed, including analysis and functional requirements. Finally, a summary of the chapter will be given in section 3.4.

3.2 System Model

In this section we will give a description about the model that we used to build up our systems. Figure 3.1, illustrates the smart grid application model using sensor WS. The model consists of smart homes connected with a web server through the internet. The utility as well as the inhabitant have the ability to access the server and demand specific data. In the next part we will explain the smart home components and how it is communicated and how it works.

3.2.1 Smart Home Components

Smart home is constructed with a special network infrastructure to enable an owner to remotely control any appliance in the home. Furthermore, the future smart home has the ability to generate its need of electricity from the renewable resources like wind and solar, and to sell the extra energy back to the grid. Moreover, the utility systems integrate with the smart home energy management systems. In addition, a two-way-communication exists between them. In the following, an overview about the smart appliances, sensor nodes, network medium, and the web server is given.

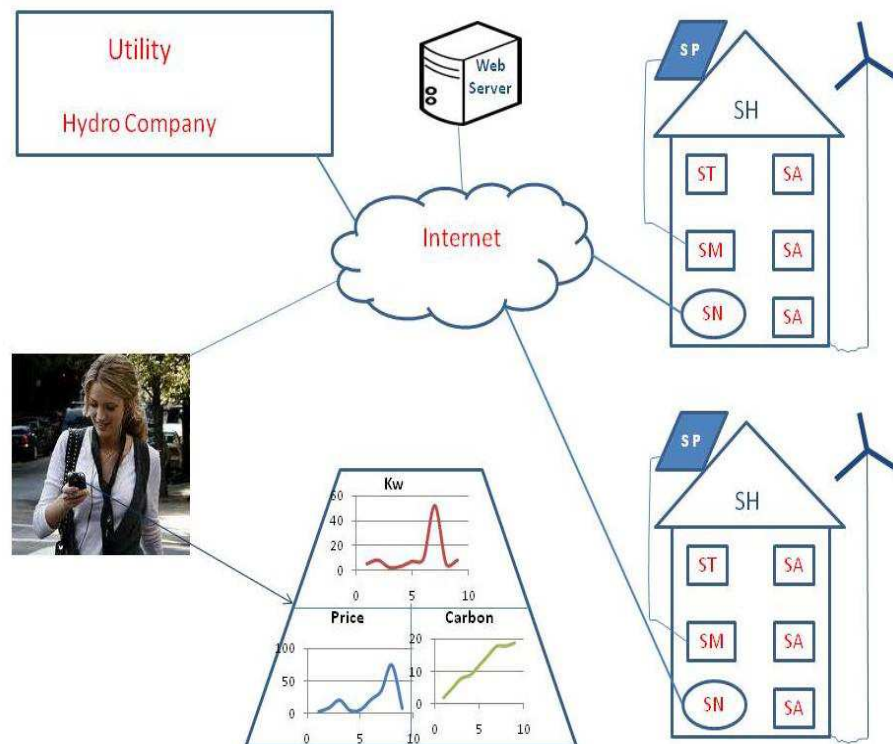


Figure 3.1: Smart grid applications model using sensor network web services

3.2.1.1 Smart Appliance

The smart appliance utilizes modern technologies and communications to provide a better function, reduce the cost of consumption and more energy-saving. In our model we have

mainly a smart Air Conditioning (AC) appliance. A sensor node is combined with the AC to measure the current energy consumption and sends it to the server and vice versa, as well as receiving data from the server to update the current AC working status.

3.2.1.2 Sensor Nodes

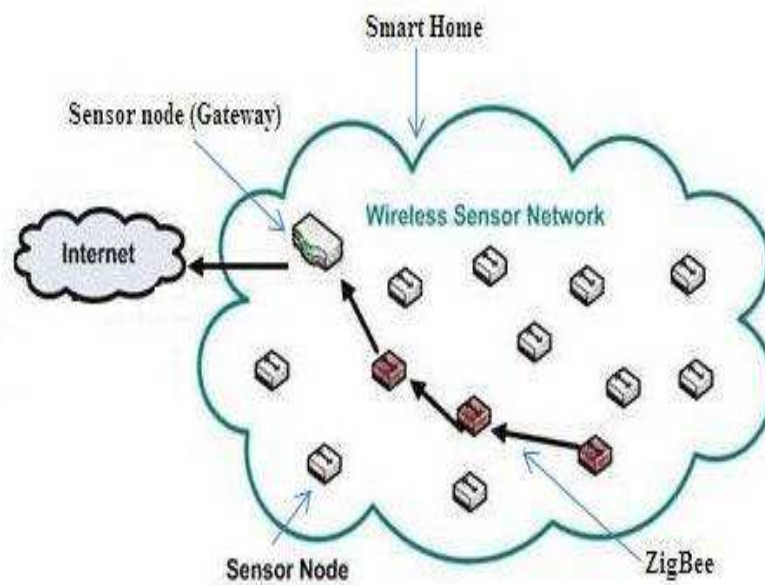


Figure 3.2: Smart home sensor wireless network

As our system architecture consists of Wireless Sensor Network(WSN), the WSN combines many sensor nodes in the smart home using a suitable communication medium. Sensor node [40] is a node in the wireless sensor network that has the ability to measure physical phenomenon, like temperature, weather humidity, and electricity consumption, process these data, and send it to another node in the network.

In our implementation the sensor network deployed in the house, mainly has sensors combined to the smart appliances, the data transfer from one node to another to reach

the gateway sensor node and vice versa. The main node acts as a network gateway as shown in Figure 3.2, which connects the house sensor network with the web server through the internet using ZigBee protocol.

3.2.1.3 Network Medium

In order for the sensor nodes to communicate with each other, a ZigBee protocol is used. ZigBee [41] is a high level communication protocol, using digital radios for wireless personal area network, such as electrical meters, wireless light switches with lamps. In addition ZigBee has many advantages [42] to use in our design:

- 1 : Low cost: Since the smart home is going to be the future home, we are looking for cheap and efficient technology to deploy in the future homes, in order to enable the inhabitant with different incomes to use it.
- 2 : Reliability: Zigbee MAC layer deploy listen before send mechanism, so before the node start sending the data, it makes sure that the current channel is free, then it starts transmit each packet, and waiting for the receiver confirmation. If no confirmation received, the node will transmit the packet again.
- 3 : Low-power: ZigBee is an ideal solution for the smart home since it is an energy efficient protocol. ZigBee node has a low transmission rate, in addition, the amount of data transmission is small, therefore, the time of receiving and sending signal is reduced. On the other hand, the node on the off-mode is sleeping and does not spend energy.
- 4 : expansion flexibility: A zigBee network is able to handle 255 nodes including one master and the others are all slaves. In addition it is easy to add extra nodes to the already installed Zigbee network without redesign the existing one.
- 5 : Many Network topologies support: like mesh, star, beer to beer. In our Sensor Network design, we have a gateway node communicating with each other nodes

directly, composing a star topology which is supported by ZigBee protocol.

5 : Security: Zigbee had a mechanism to check the data integrity and sender authentication.

6 : Low latency devices support [43]: The Zigbee node does not have to synchronize with other nodes before start sending. So, the device needs only 30 milliseconds to join the network and 15 millisecond to start communicating and sending data. This makes ZigBee appropriate technology to use in the smart homes.

3.2.1.4 Web Server

A web server is a computer program used to deliver the server contents to the requester (web client) most likely in HTML format using Hypertext Transfer Protocol (HTTP). The client or the web browser initiates the request for a specific resources on the server side through the internet connection, and the web server response back to the client as an HTML document and any additional contents like images, Cascade Style Sheets(CSS), and java scripts.

In our design the web server resides on the utility, and it communicates with the house sensor node through the internet. On the the other hand, the home owner has the ability to access the web server and gets certain information about his/her home energy behavior. In addition, a database server is installed to store all the information collected from the applications. The database server is located at the utility and it is connected with the web server.

3.3 System Analysis

In this section we will describe the applications that were developed. The first application is reading the home meter remotely for billing, where the utility server gets the current meter reading and stores it in the database server with the current time for the billing

and statistics purpose. The second application is Remote Appliances Control, in this application the server has the ability to control the home appliance electricity usage at the peak hours. The last application is Managing Smart Homes Renewable Energy. Since the home uses the renewable energy has a two main source of the electricity, this application will control the usage of the house energy from both sources and the best time to use each one.

3.3.1 Applications Description

3.3.1.1 Reading the home meter remotely for billing

The traditional meter reading needs the utility agent to visit each unit in the grid and reads the current meter value and register it on the paper. After that, another data entry clerk has to feed the utility billing system with these values. This way has many drawbacks:

- 1 : Time Consuming: since the grid contains thousands of units, the time taken to cover all the neighborhood and read the current meter values is time wasting and needs several days to finish it.
- 2 : Resource intensive: a lot of human resources are involved in the operation, some of them are moving to read the values and others are employed to feed the utility system with the data.
- 3 :Inaccurate: The clerk who reads the current meter or the data entry employee may do some mistakes while working and that would affect the accuracy of the data provided then the customer bill.

Therefore, the idea to make an automatic system to read the meter reading, store the data, and generate the customer monthly bill becomes important for the utility as well as the customer. This system would be a part of the utility management system

and it should have the ability to communicate with other utility subsystems, like Meter Data Management (MDM), Customer Care and Billing Systems (CIS) and Outage Management Systems (OMS) [9].

The idea we are presenting is similar in terms of functionality to the smart meter system, which are installed in some regions. However, in our implementation we have a sensor node combined to the smart meter that has the ability to communicate with the utility server and propagates the data using the WS technology.

Reading the home meter remotely application needs the web server to access the sensor node to get the current meter reading. The value obtained and the time stamp, will be stored in the database server. Such system would save time since all the unit meters could be read at one time, and there is no need for the utility clerks or for the data entry section. Moreover, this application allows to perform Interval measurement, where the system can get a certain meter reading value several times in different periods, in order to measure the home consumption during a period of time. The interval measurement will be a valuable information to the home owner to track his/her home electricity consumption and to take the appropriate action if needed. In addition, the grid operator would have a periodic report about a certain neighborhood consumption to optimize the provided service and control the load.

3.3.1.2 Remote Appliances Control

As the electricity demand is rising, the utility has to arrange its generators in order to respond the high demands. The high demand normally happens in some periods of the day called as peak hours. Figure 3.3 represents the demand period distributed during one day, it can be seen that the electricity consumption is different from time to another during the day according to the human activities. For example in winter, peak hours last 8 hours a day, one peak in the morning from 7:00 am to 11:00 am and the another at the evening from 5:00 pm to 9:00 pm[6].

The term Demand Response refers to the mechanism of managing the customer con-

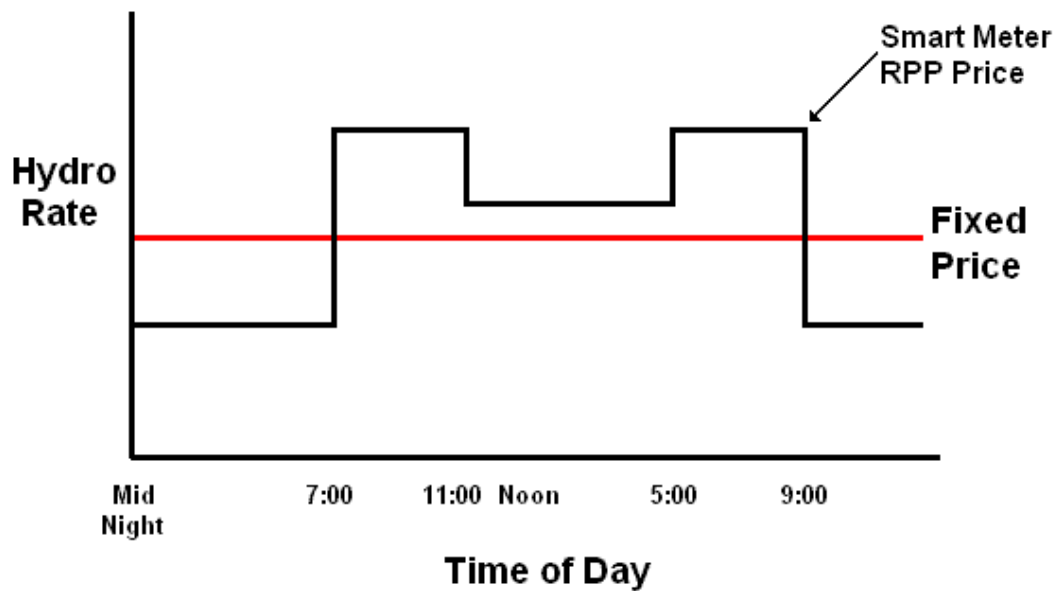


Figure 3.3: Constant vs. Fluctuating Electricity Rates in Ontario [6]

sumption in response to the supplier condition. Therefore, the utilities incentivize the customers to reduce their electricity consumption during the peak hours, in order to do so, some promotions were announced by the utilities to encourage the customer to shift their usage to the off peak periods. Ontario Hydro [6], sets different charging rates during the day, the dynamic charging rates using the smart meter to assign the price during the day time. The highest rate is 9.1 cents/kilowatt hour (kWh) which affects during the peak hours, and the off peak hours charges is 4.2 cents/kWh.

On the other hand, the utility can control the appliances at the peak hours based on previous agreement with the inhabitants. This enables the grid to work properly during the peak hours by decreasing the amount of residential load.

One of the high usage consuming appliance is the Air Condition (AC). If all the inhabitants turn their AC's on the hot days, that would cause an overload demand on the utility generator side, and the utility will not be able to provide its services properly. Therefore, the system should manage the AC usage during peak hours, and decreases

its consumption if the overall demand increases above a certain level. The utility server communicates with the sensor combined to the AC and gets the current consumption. Depending on the algorithm, the system determines if the AC has to decrease its usage or stay the same. Furthermore, the sensor has to communicate with several homes AC to take the appropriate action.

3.3.1.3 Managing Smart Homes Renewable Energy

Integration of renewable resources are considered as one of the most important goals of the smart grid. Energy generated by means of solar power, wind power, etc., can be stored for future use by the home appliances or it can be sold to the grid. Many leading companies are working to install solar panels and wind turbines in the homes. In the mean while, the renewable energy market is growing rapidly in the last few years, for instance, the wind power generator is growing at the rate of 30% annually, with a total world wide installing capacity of 158 GigaWatts in 2009[44].

Many factors pushes the world to think about the renewable resources, one is the the climate change and the global warming due to the human activities, thats caused by burning fossil fuels which increases green house gases. In addition to the high oil prices and the peak oil where the oil production in the world is reached its maximum point. Figure 3.4, illustrates that the decline phase has started and the oil fuel is decreasing. Deploying the renewable energy resources, mainly the wind turbine and the solar panel, have started in the fifties last century[9]. The production of electricity by using solar and wind have many advantages, first its clean source of energy which does not produce any waste gases or cause pollution, in addition once the equipment is installed, the production is supposed to be free, the wind and the solar are readily available for everybody.

Nowadays, Photovoltaic (PV) solar panel and wind turbine, are ubiquitous in marketing campaigns, in addition, many newly constructed houses are considered to install the PV solar Panel or wind turbine, with encouragements and promotions from the electricity utilities and some private sectors dealing with the installation of the renewable

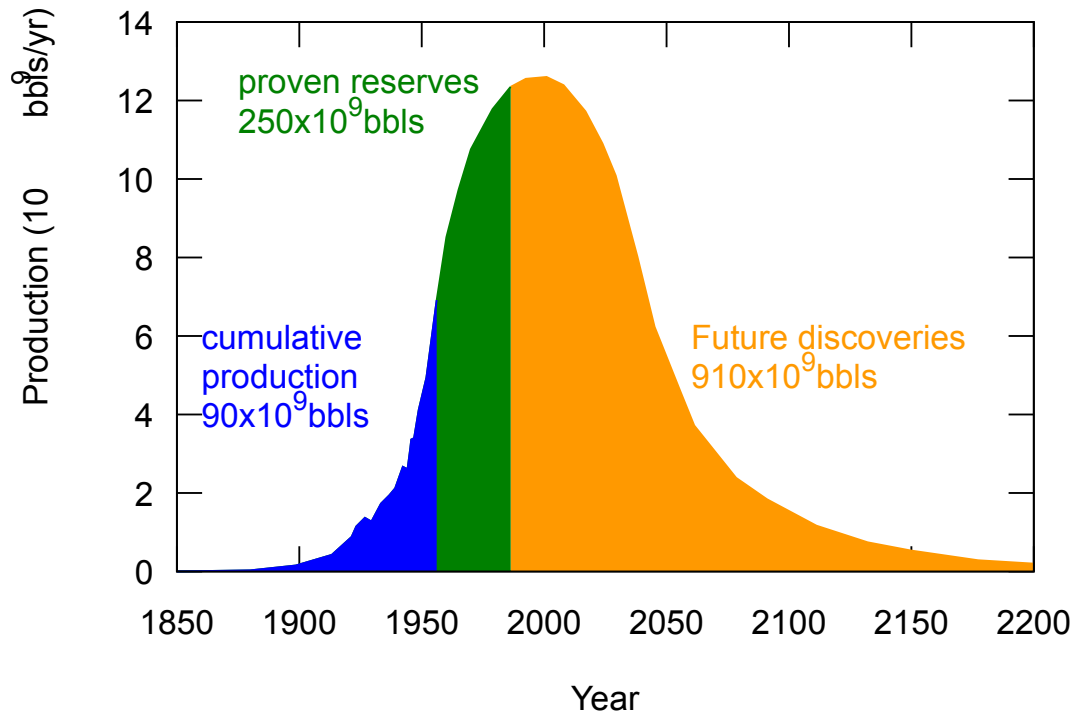


Figure 3.4: A logistic distribution shaped for Oil production curve [7]

equipments.

Therefore, the rapid spread of the renewable technology, and the increasing numbers of the leading companies focusing on the installation and the maintenance of the PV solar panel and wind turbine, ask for the need of an application to manage the integration of the renewable energy with the current electricity grid.

The home with the renewable energy resources may use the energy from the utility, or the energy from the renewable resources, or it could sell the electricity back to the grid when the home has extra energy from the resources. Furthermore, there is a question about the ideal time the inhabitant should use the utility electricity or the renewable energy or sell back the electricity.

The application should manage the usage and energy flow, in addition, it should keep track of the smart home energy usage performance. So, any management application must consider three cases:

- 1 : The home is using electricity from the grid even it might have energy on the storage devices.
- 2 : The home quits using electricity from the grid and uses locally generated or stored energy.
- 3 : The home sells electricity to the grid.

The user may choose to do one of the above actions depending on the price of electricity and the amount of energy stored. The utility applies rates that vary with the time of day which is called Time Of Use(TOU) billing.

3.4 Summary

In this chapter we presented the design and architecture of the network model and the network technologies we used in our applications. In addition we analyzed the system we developed and its characteristics.

The system model used to develop the applications consists of smart homes connected to each other. Every unit, has a wireless sensor network, each sensor node is combined with an appliance. The WSN is connected via ZigBee protocol. The home is connected with the utility server through a sensor node via a WS that acts as a gateway. Furthermore, the gateway node communicates with the other sensor nodes in the home via a WS as well, and it has the ability to send and receive the data. The inhabitant could access the web server and gets specific data depending on a pre-privileges granted.

On the other hand, three smart grid applications were introduced. The first application is reading the home meter remotely, which enables the utility to access the meter and gets the reading and registers it with the current time on its database. The second application is Remote Appliances Control, where the utility server could manage the usage of the home appliance electricity during peak hours. The third application is Managing Smart Homes Renewable Energy, where the home consider the renewable

energy production. This application manages the flow of the electricity from either the renewable or the grid and it decides when to use each one.

Chapter 4

System Design and Implementation

4.1 Introduction

In this Chapter we will show the applications implementation and the system diagrams. The three applications are: Smart meter reading remotely, Remotely control appliances, and Managing the renewable energy. For each application we will model the system dynamics, focusing on behavioral and interaction diagrams, furthermore, the algorithm for each application will be introduced.

In the next section we will give the sequence diagrams for the applications which illustrate a detailed scenario of the system executions. To show the system behavioral aspect, the interaction diagrams will be presented in Section 4.5. Next section will present the algorithms and discussed the process flowchart of each application. The last section is the conclusion.

4.2 Application Implementation

The applications were implemented using Java NetBeans Framework 6.8 [45], and MySQL open source database engine server [46]. Java NetBeans is a flexible tool to build a WS applications, it has a complete WS stack called METRO [47]. Metro enables the

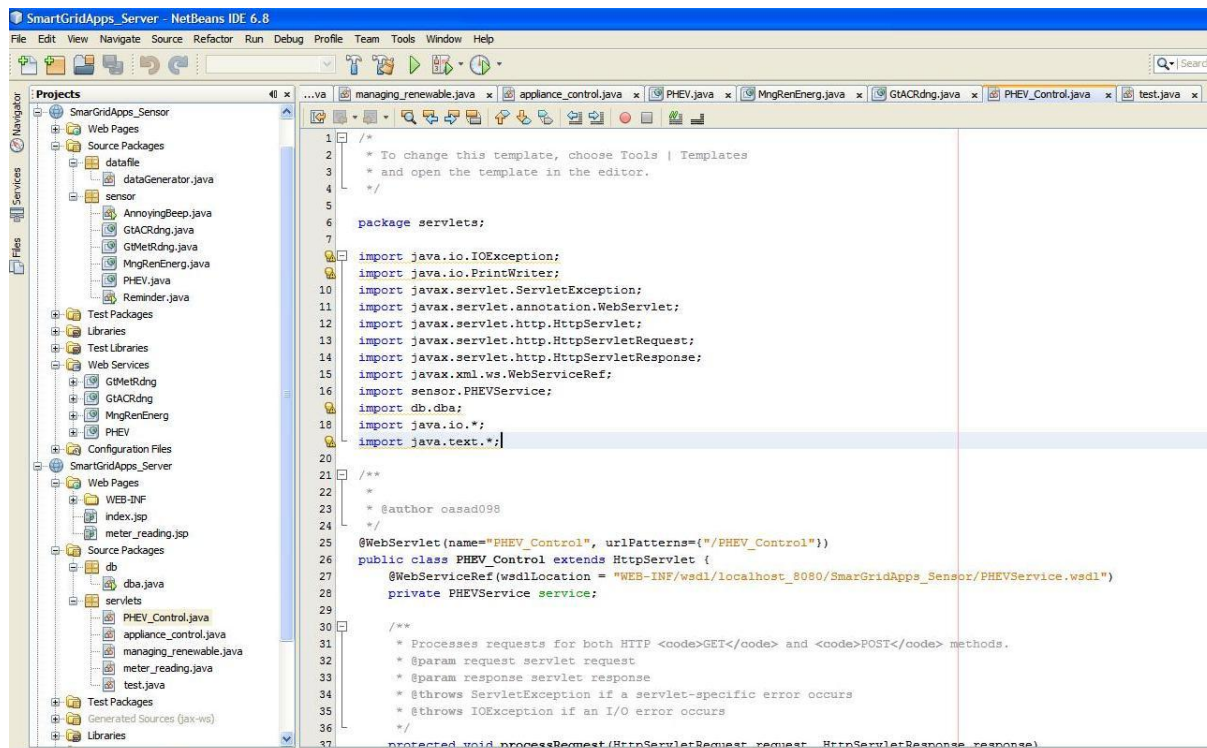


Figure 4.1: Java NetBeans Framework for Sensor Web Services Applications

developers to create and implement secure, reliable, transactional, interoperable WS in addition to the client. Furthermore, Metro can support the two common standard WS Protocols: SOAP and REST.

Since the system were implemented on a single machine, we assume that there are two separate projects nodes, the first one is *SmartGridApps_Sensor* represents the sensor which runs the WS, and the other is *SmartGridApps_Server* for the web server. Figure 4.1 is the screen of the NetBeans IDE that illustrates the two project nodes we use.

The web server node contains two packages, one is the *servlet* package that contains all the web server codes which communicates with the sensor node via the WS. Each application is represented with one file that runs the algorithm. In addition, the *db* package has the files responsible to manipulate the database server.

On the other hand, the *SmartGridApps_Sensor* node represents the sensor nodes, and

it runs the WS that can communicate with the appliances to get the desired data. We used two packages on the sensor project side, depending on the system architecture that was discussed on the previous chapter, the first one is the *Sensor* package, which has all the WS needed for the applications. The other package, is the *datafile* which acts like a data generator file used to generate the values requested by the WS on the *Sensor* package.

4.2.1 Data generator file

The system design relies on a web server communicating with a home sensor network through a gateway sensor node, each sensor node has the ability to communicate and to sense the data necessary for the applications, as well as receive data from the server. Hence, every sensor node has to generate realistic information and send it to the web server to process it and get the result. Therefore, the data generator file was implemented to act like a sensor node.

The data generator file used to act as three sensor nodes:

- 1 : The sensor node combined with the meter: so it can generate meter readings in kWh and send it to the web server through the gateway node.
- 2 : The sensor node combined with the AC: it has the ability to generate random numbers represent the current AC consumption.
- 3 : The sensor node combined with the storage device: in this case the data generator file has to produce a number which represents the amount of energy that exists on the storage during a period of time.

The WS located on the gateway sensor node could access this file and read the numbers related to each application in order to send it to the web server.

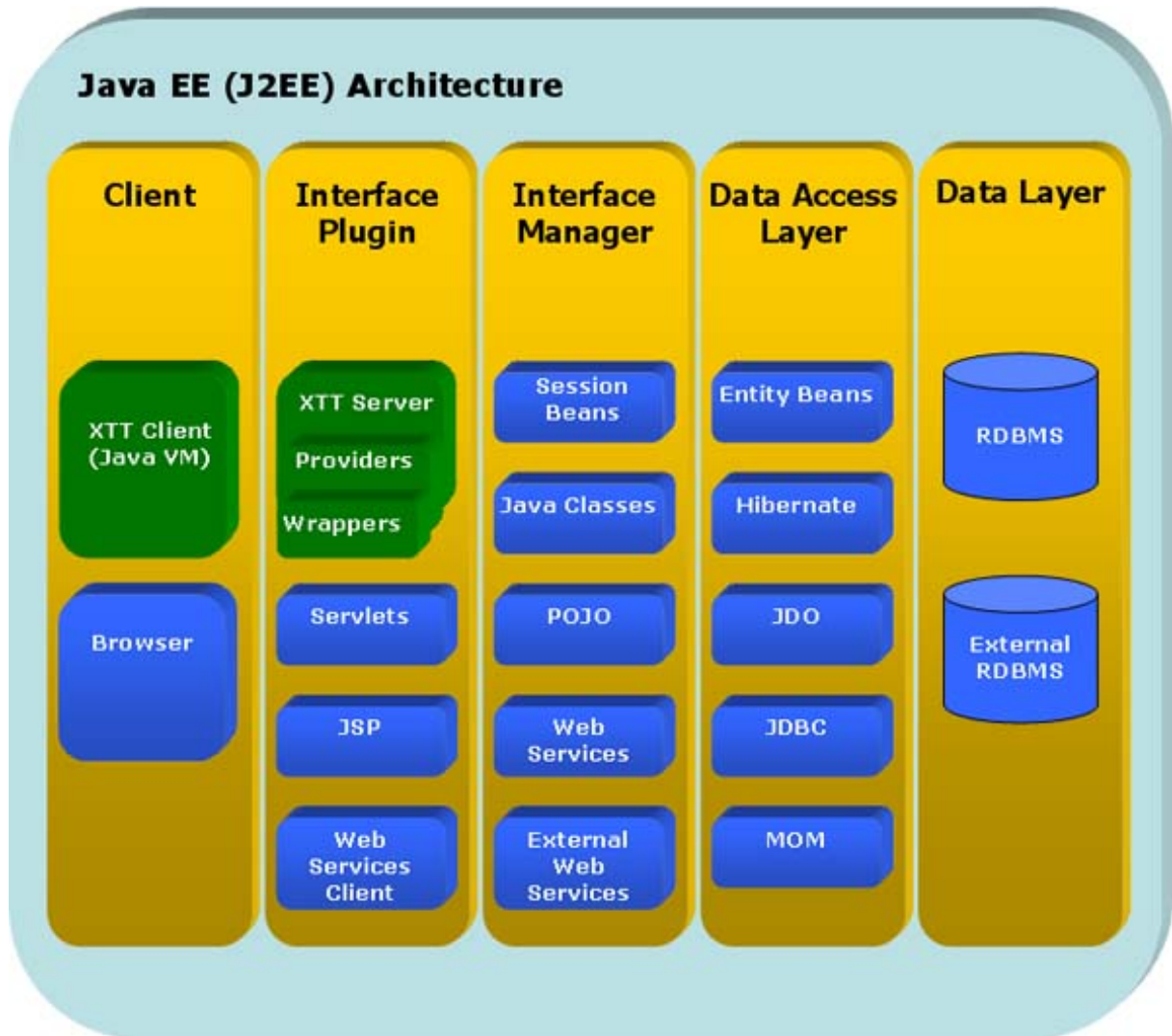


Figure 4.2: Java NetBeans Framework for Sensor Web Services Applications [8]

4.3 J2EE Architecture

Java 2 Enterprise Edition (J2EE) defines the standards to develop a multi tier application in java. Figure 4.2 illustrates the Java NetBeans framework for sensor WS applications. The framework simplifies the enterprise applications basing them upon standardized components [48].

In our implementation we track the J2EE standards to build up the application,

those standards can be applied to our system since it is a multi tier distributed model. In addition, it will be easy to integrate with other enterprises applications.

The three common tiers, which have been used on the implementation are:

- 1 : Client Tier: represented by the home owner web browser or the utility clerk who has access to the system.
- 2 : Application Server Tier: this tier hosts the presentation and business logic. The presentation logic run by the web server servlets and it sends the response back to the client in the proper format. On the other hand, the business logic is presented by the algorithms on both the web server and the sensor node .
- 3 : Back end Tier: This tier is the data source which access the database and stores the required information or request specific data from the data base server.

4.4 Data Base design and Tables

To store the information generated from the applications, in order for the end user as well as the utility to access it, MySql data base management system was used. Figure 4.3 illustrates the tables used to save the data.

4.4.1 Database Tables

Eventually, we build up the data base scheme depending the needs to store the values and get the results. Therefore the first table is the *homes* table, which stores all the information related to end point units like the id, owner, address, telephone, and the date of subscription. This table is used so that other tables can store the information related to a specific home.

For the first application, reading meter remotely, table *meter_reading* is used, which stores the meter readings output from the application and the time stamp when the reading was taken.

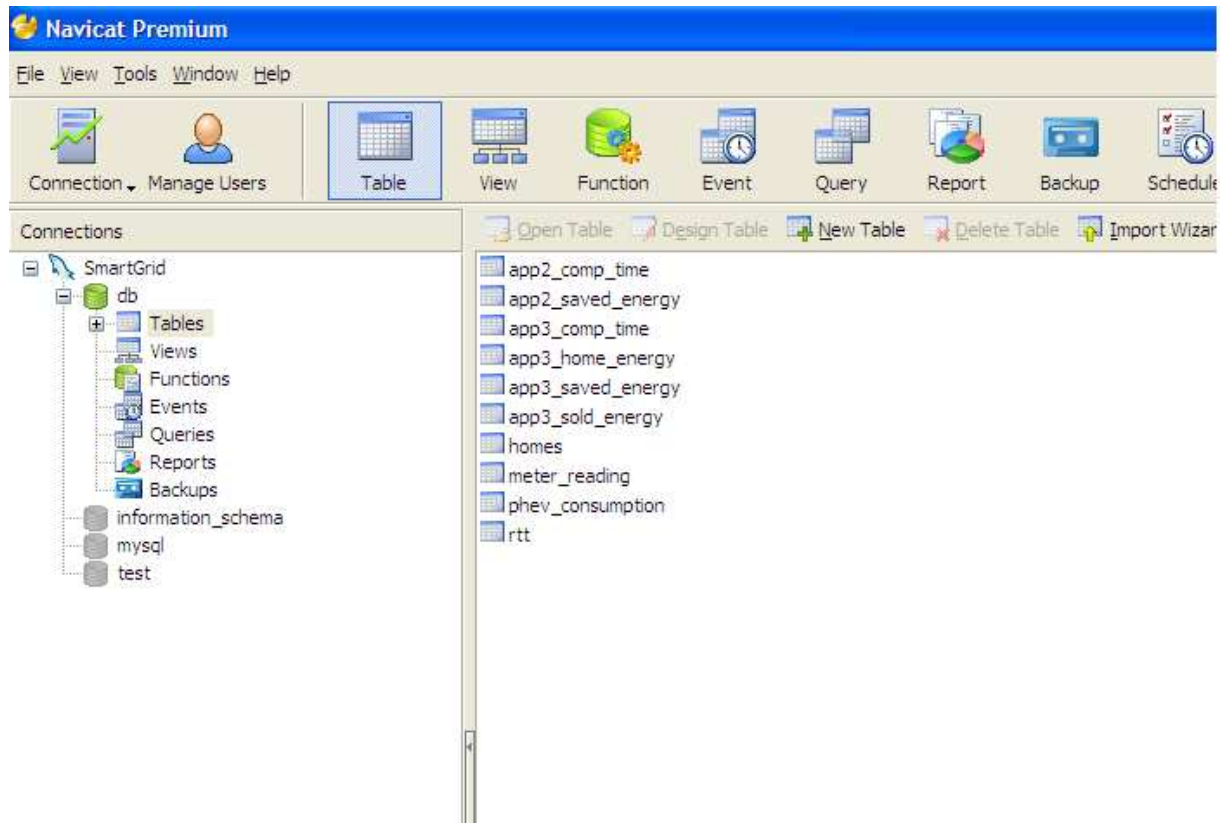


Figure 4.3: Mysql Data base tables used for the applications

To save the data generated from the second application, Remote Appliances Control, table *app2_saved_energy* was used. The table stores the amount of energy saved by a specific home AC when running the applications and the time the saving takes place.

In order to save the information for Managing Renewable Energy application, table *app3_saved_energy* is used, it stores the amount of energy used for a house by the current, and the amount of electricity used from the storage device, and the amount of energy sold back to the grid.

The other tables are created to record the completion time for each application, which defines how much time it takes to get back the data from the server and complete the process. Next chapter presents, in details, the completion time for each application.

4.4.1.1 Database Relations

The table *homes* contains all the information about every home belongs to the system with the primary key *home_id*. And every other table has its unique primary key. Therefore, the other tables store the *home_id* and use it as a foreign key to keep track with the home.

4.4.2 JDBC Connector

Java Database Connectivity (JDBC) is a java API provides connectivity between the the java programming language and the database management system. JDBC makes three things possible: [49]

- 1 : Create a connection with the database and access the database sources.
- 2 : Send SQL statements.
- 3 : Receive the results and process it.

In our implementation we use a Connector/J to connect the java with the database. The connector provides a connectivity for cleint applications developed in java programming languages through a JDBC driver [50]. Figure 4.4 shows the JDBC connection string used to connect to MySQL.

4.5 Interaction Diagrams

Interaction diagram represents the model of the dynamic system components, and gives a better view of how the system is working [51]. The diagrams consist of a set of actors and objects communicate with each other to perform the system functionality.

On the other hand, the Interaction diagram shows several types of communication, like message exchanges over a network, and the command issued by the end user. The

```

* @author omar
*/
public class dba { // this class is the data base agents that talks directly to the database and it has all the quires :
    static Connection conn;
    public dba() // the constructor which established the db connection
    {
        try {
            String connString = "jdbc:mysql://localhost:3306/db";
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            conn = DriverManager.getConnection(connString, "root", "root");
        } catch (Exception e) {
            e.printStackTrace();
            System.err.println ("Cannot connect to database server");
        }
    }
}

```

Figure 4.4: JDBC connector to MySql from Java

elements that can be found in the interaction diagrams include: Instance of the class or actors and the messages.

4.5.1 Sequence Diagram

One of the common form of the Interaction diagrams is the Sequence diagrams, it represents the message exchanged by set of objects or actors performing a specific task [51].

In the next subsections, we will show the sequence diagrams for each application, and presenting how messages exchange in each application between the objects located on the sensor node and the web server with respect to time.

4.5.1.1 Reading Meter Remotely Design Sequence Diagram

Figure 4.5 shows how the class are working together to allow getting the Meter Reading, each object interacts with the other objects when the operation is begun by the user, till saving the data in the DB server. The figure include one actor, the end user, and four objects, *MeterReading*, *GtMetRdng*, *DataGenerator*, and *DB*.

The process starts when the end user asks the *MeterReading* interface located on

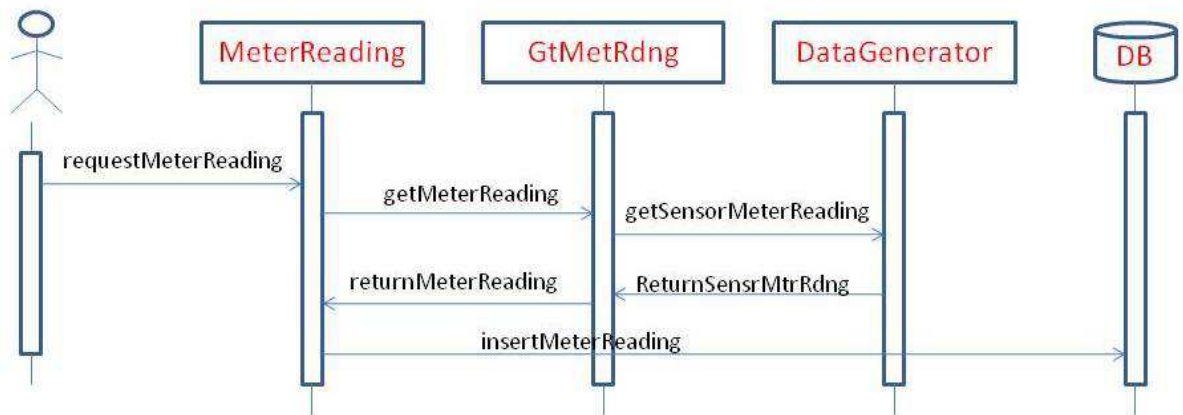


Figure 4.5: Sequence diagram showing the process of Reading Meter Remotely

the web server to get the current meter reading. Then, the message *getMeterReading* generated by the web server is sent to the *GtMetRdng* object located on the gateway sensor node, after that, another message *getSensorMeterReading* is sent to the sensor node combined with the meter to ask for the current meter reading. However, in the implementation, the sensor node combined with the meter is represented by a *DataGenerator* object, which acts the same as the sensor node, and has the ability to generate a reading numbers.

Once the *GtMetRdng* object receives the *returnSensrMtrRdng* message, it forwards the data to the *MeterReading* object on the web server. Then, the *MeterReading* object sends the current meter reading to the *DB* objects to store the data on the BD server.

From the figure, the process time line can be traced, since the process happened first is located above the one happened next.

4.5.1.2 Remote Appliances Control Sequence Diagram

Figure 4.6, represents the sequence diagram for Remote appliances control. As the previous application, The actor initiates the interaction via the user interface, the user

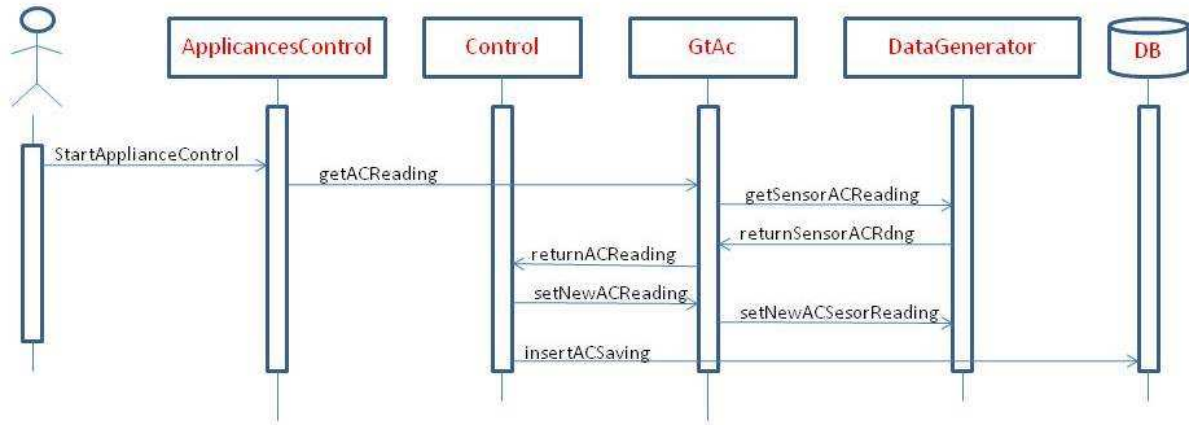


Figure 4.6: Sequence diagram showing the process of Remote Appliances Control

interface sends a *StartApplianceControl* message to the *AppliancesControl* objects located on the web server. Once the web server receives the message, it generates a *getACReading* to the *GtAc* located on the gateway sensor. After that, another message *getSensorACReading* generated by the *GtAc* object sends to the sensor node combined to the AC to ask about the current AC reading. In the our implemetation, the sensor node combined with the AC is represented by *DataGenerator* which can produce an AC readings like a sensor.

The AC sensor node responds to the request by sending back the message *returnSensorACRdng* containing the current AC reading. When the *GtAc* object located on the gateway sensor receives the message, it propagates the *returnACReading* message to *Control* object located on the web server. The web server decides wether the AC has to be reset to a specific value depends on an algorithm it has. If yes, then, it sends a new message *setNewACReading* to the *GtAc* located on a gateway sensor node, in addition to another message *insertACSaving* to the *DB* object to store the AC saving values on the data base server. Once the gateway receives the new value, it includes it in the message *setNewACSesorReading*, and sends it to the sensor node combined with the AC. The AC

decreases its current consumption depending on the message it receives.

4.5.1.3 Managing Smart Homes Renewable Energy Sequence Diagram

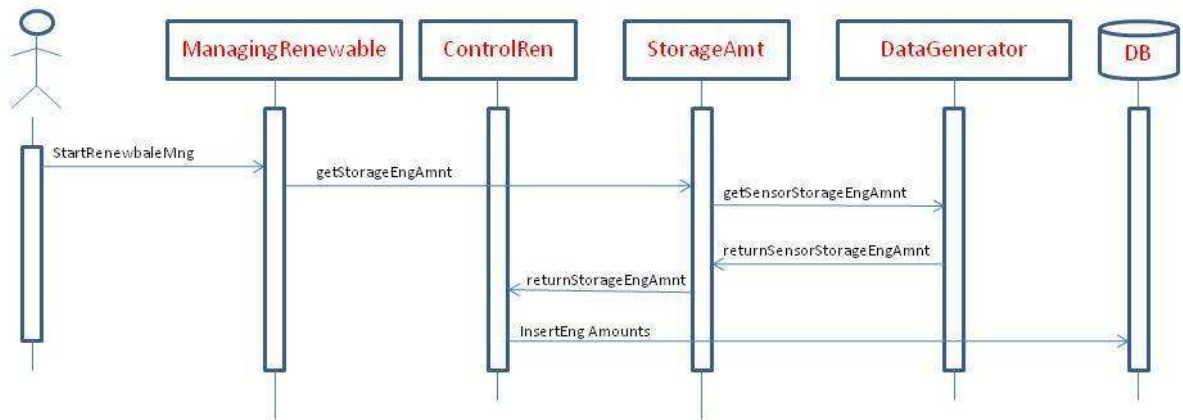


Figure 4.7: Sequence diagram showing the process of Managing Smart Homes Renewable Energy

The sequence diagram for managing smart homes renewable energy is represented by Figure 4.8. The application starts once the end user initiates the interaction via the user interface. The user interface in the diagram is represented by *ManagingRenewable* object. The object sends a request message *getStorageEngAmnt* to the *StorageAmt* object located on the gateway sensor. Then, the gateway sensor object *StorageAmt*, generates a *getSensorStorageEngAmnt* message to the object located on the sensor node combined with the storage device on the home, in our implementation the object resides on the sensor node that is represented by the *DataGenerator*.

Then the *DataGenerator* sends a message response *returnSensorStorageEngAmnt* to the *StorageAmt* object. Then, the gateway object propagates another message *returnStorageEngAmnt* to the *ControlRen* resides on the web server. The *ControlRen* runs an algorithm to decide the suitable action to take. In addition, it sends the information in a message *InsertEngAmounts* to the DB object to save the saving amounts.

4.5.2 Activity diagram

Activity diagram represents the business process and the work flow in the system, therefore, it is a dynamic diagram that shows the different events in the system and the event that cause a an object to be in a particular state [51]. In the following parts, we will present the activity diagram for each application and describe it.

4.5.2.1 Reading Meter Remotely Design Activity Diagram

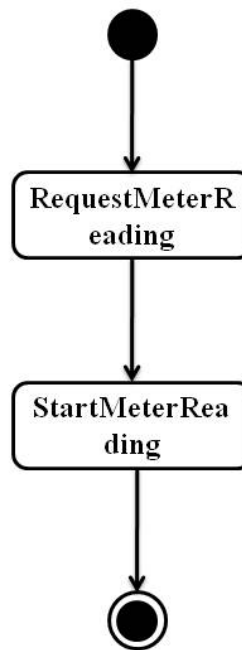


Figure 4.8: Activity diagram showing the process of Reading Meter Remotely

The activity diagram starts when the end user initiates a *RequestMeterReading* process, then, that process generates another process *StartMeterReading* which can get the

current meter reading. After the final process is completed, the application stops.

4.5.2.2 Remote Appliances Control Activity Diagram

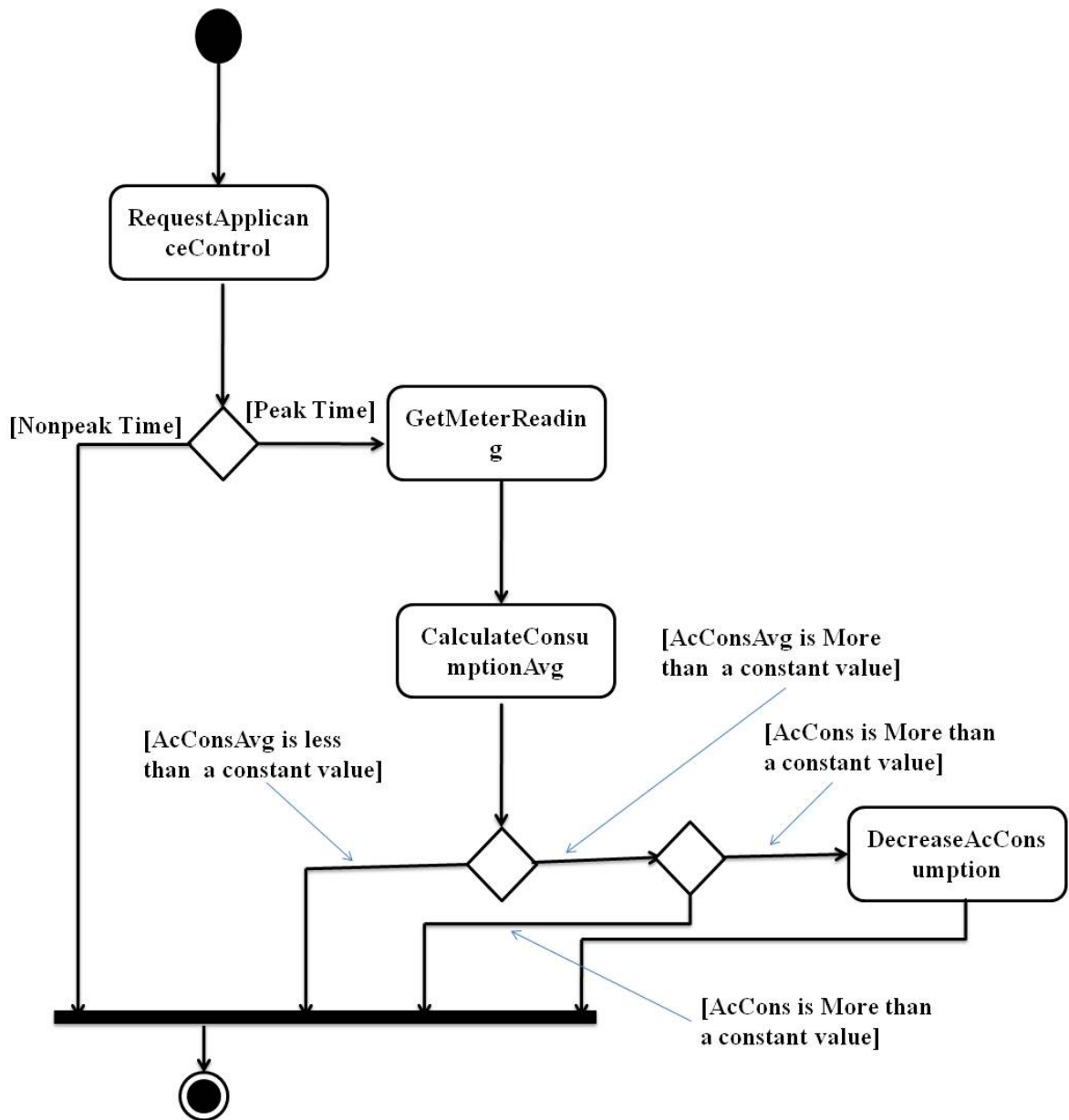


Figure 4.9: Sequence diagram showing the process of Remote Appliances Control Activity

Figure 4.9, represents the activity diagram for the application Remote Appliances Control. The activity diagram starts when the end user requests the appliance control application. Then, the application has to take the decision as to run or not depending on the current time, if the time is not within the pre-defined peak hours, the application terminates, otherwise, it continues. The application asks to get the current AC reading from the sensor node combined with the AC. After that, the server calculates the consumption average when receiving all the ACs in a specific area. The decision will be made depending on a pre-defined constant value, if the average is below that value, the application terminates. Otherwise, the application will ask every AC that has a consumption exceeding the average value to lower its value so as to meet it. At the end of the application, all the ACs that are consuming more than the average value will be decreased back to the average.

4.5.2.3 Managing Smart Homes Renewable Energy Activity Diagram

Figure 4.10, represents the activity diagram for the application Managing Smart Homes Renewable Energy. The activity diagram starts when the end user initiates a request message from the interface, as the previous application, the system checks whether the current time within the peak periods or not, if it is not the application terminates. Otherwise, the system asks to get the storage amount from the sensor node combined with the storage device. If that amount is less than a certain value, defined as the amount of electricity needed for one day home use, then the amount on the system ask to use that amount for the home needs. Otherwise, if that amount is more than the home needs for one day, the process continues as follows: one process is initiated to use the amount of electricity for one day, and another process is formed to sell the extra amount to the grid. Finally, the application terminates.

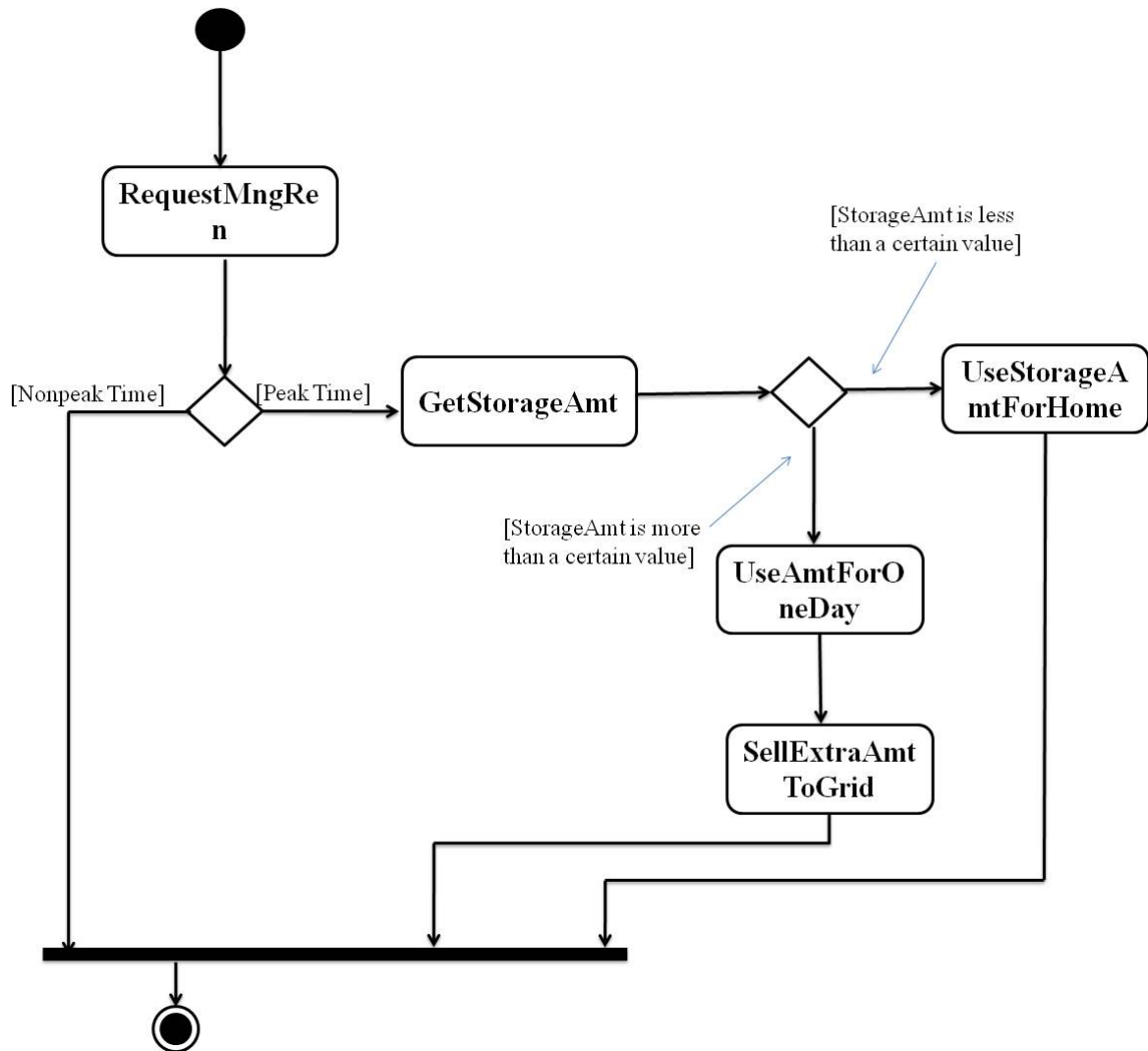


Figure 4.10: Activity diagram showing the process of Managing Smart Homes Renewable Energy

4.6 Applications Algorithms

The algorithm defined as an effective method to solve a problem expressed in a finite sequence of steps. So the algorithms should perform in specific order to carry out a specific task. Thus, the algorithm can be considered to be a sequence of operation which

could be simulated using Turing machine.

Since our applications are computer driven programs, each one of them has a set of tasks that can be presented as an algorithm. In the following subsections, the algorithms for all the applications would be presented and discussed.

4.6.0.4 Reading Meter Remotely Algorithm

The idea of the Reading Meter Remotely application is that the web server communicates with the sensor gateway in the house in order to get the current meter reading. Then the gateway sensor accesses the WS located on the sensor node combined with the meter, which could get the meter reading from the meter. The meter reading which is taken from the sensor node combined with meter is assigned to a variable in the web server.

The web server connected to database server, thus, the values of the current meter reading are stored in a specific table with the current time stamp. The utility could use the information to generate the monthly bill for the inhabitants, or it could use it to get the home consumption over different time intervals. In addition, the inhabitants have the ability to access the database server based on a previous agreement to watch his/her home electricity behavior.

4.6.1 Remote Appliances Control Algorithm

The algorithm of Remote Appliances Control, Algorithm1, is responsible to control the home appliances during peak hours, and keep the utility generators in a good condition in terms of energy supply when the peak happens. The algorithm starts by checking whether the current time T is located within the peak hours P . However, the peak hours could be manually changed by the utility to fit the different peak periods within the year. If the current time does not fall within the peak hours, the system algorithm stops and the system will terminate.

Otherwise, the algorithm loops to all homes in a specific neighborhood, it starts reading the current AC consumption C_i , and accumulates the result. Once all the desired

Algorithm 1 Remote Appliances Control

```

{ $C_i$ =the current energy usage for AC in home  $i$ }
{ $C_{total}$ =the total energy usage for specific AC's}
{ $C_{avg}$ =the average energy usage for specific AC's}
{ $C_{const}$ =the allowance limit of energy usage for one AC}
{ $T$ =current time}
{ $P$ =peak hours}
{ $i$ =counter}
{ $n$ =number of homes in geographic area}
if  $T \in P$  then
  for  $i = 1$  to  $n$  do
     $C_{total} \leftarrow C_{total} + C_i$ 
  end for
   $C_{avg} \leftarrow C_{total}/n$ 
  if  $C_{avg} \geq C_{const}$  then
    for  $i = 1$  to  $n$  do
      if  $C_i \geq C_{const}$  then
         $C_i \leftarrow C_{const}$ 
      end if
    end for
  end if
end if

```

homes are completed, the algorithm takes the average of all the readings. This done by dividing the addition result C_{total} by the number of the homes in the covered area.

Then the algorithm compares the average result C_{avg} with a constant C_{const} . The constant is actually the limit of the energy usage for one AC in the peak hours, and it could be changed by the utility depending on how much load is being put on the generators. If the average is less than the constant value, then nothing happens and

the homes AC's continue to work, and the algorithm will terminate. Otherwise, if the average is bigger than the constant value, then the algorithm has to loop again for all the AC's consumption values that was read on a previous step. Then, it checks every AC, if it has a bigger consumption value than the constant value, it asks the AC to decrease it to the constant limit, otherwise, if the AC consumption value is less than or equal to the constant one, it keeps it without any change, and the AC continues to work. Once all the homes are done, the algorithm will stop.

By implementing this algorithm on the utility side, the generators will work properly during peak hours, and it will not suffer from the high demand in a certain period, Furthermore, this application could achieve a high power saving as we will show in the results section.

4.6.2 Managing Smart Homes Renewable Energy Algorithm

In this algorithm we consider a smart home that integrates renewable resources, mainly a wind turbine and a solar panel, we assume the system has a storage device combined with the sensor node to get the current saved energy amount. The sensor node carry the flow management system to coordinate when to use the stored energy, furthermore it decides whether to sell electricity back to the grid or not, depending on the amount of energy already exists on the storage device. This system is fully automated, since the inhabitant might be out of the home and there is still electricity exists in the storage device, so in this case the system could work independently and benefit from the energy without waiting for the user interaction. However the inhabitant might configure the system and change its default configuration by assigning the peak hours and amount of energy the house needs every day.

In the default case, energy is supplied from the grid. When the renewable energy is stored in adequate amounts then energy can be supplied from the storage device. On the other hand, when there is excess energy stored, the user can sell electricity to the utility. The selection of the energy supply and the decision whether to consume or to

Algorithm 2 Managing Renewable Energy

```

{ $E$ =energy amount from the storage}
{ $H$ =energy amount the home needs per day}
{ $O$ =energy amount exceeds the home needs for one day}
{ $T$ =current time}
{ $P$ =peak hours}
{ $S$ =sold Energy to the grid}
{ $U$ =used energy from the storage}
if  $T \in P$  then
   $O \leftarrow E - H$ 
  if  $O \geq 0$  then
     $S \leftarrow O$ 
     $E \leftarrow H$ 
  end if
   $U \leftarrow E$ 
end if

```

sell the energy to the grid can be controlled by the application illustrate in Algorithm 2.

The algorithm starts by asking the current time T , if it does not belong to the peak hours P already configured by the home owner, then the algorithm terminates, and the home keeps using the grid energy. Otherwise, the algorithm gets the amount of energy from the storage E that exceeds the home needs for one day H . This is done by subtracting the stored energy from the home needs for one day. If the result O is greater than zero, it is sold back to the grid and the home starts using what it needs from the storage device. Otherwise the home does not sell any amount back, and stops using the grid power, in order to let the storage starts supplying the home with the stored energy.

The described system works in the storage device sensor node in the smart home to manage the usage of the electricity flow. Furthermore, the utility integrates this application to its customer management system, in order to take the amount of the

renewable energy for the house benefit. This information can be used to provide better rates for the customer who employ more renewable energy.

4.6.3 Summary

In this chapter we talked about system design and implementation for three applications that were developed: Smart meter reading remotely, Remotely control appliances, and Managing the renewable energy.

The system was implemented using java Net beans frame work and Mysql database management system, The JDBC connector/j was used to connect the Java with MySQL. In addition, the data generator file is used to act as sensor node, and generate the numbers necessary for the applications to work.

The interaction diagram gives a better view of how the system is working by presenting the model of the dynamic system components. Two types of the interaction diagrams were presented, first is the Sequence diagram which represents the message exchange by the set of objects or actors performing a specific task, and the second is the Activity diagram which is defined as a dynamic diagram that shows the different events in the system and the event that causes an object to be in a particular state. At the end, the algorithm for each application is given, showing the processes and the events flowing for each one.

Chapter 5

Results and System Performance

5.1 Introduction

This chapter presents the results of the applications and evaluate the performance. The results give a better understanding of the applications and help the evaluation of the applications based on several metrics.

Also, the system is tested on a single PC acting like both the web server and sensor node. However, we take the sensor node characteristics in consideration when implementing the sensor side code. Hence, we assume the Pacemate[2] sensor node hardware platform, which has a 32 KB RAM, and compare the sensor code and messages with this sensor capabilities.

In addition, the system was developed in java NetBeans 6.8 which has a built in SOAP web services, and MySql server 5.1 was used. We test the application on a PC running the Windows operating system with the following characteristics: Core 2 Due CPU, 2.99 GHz CPU, and 3 GB of RAM.

In the next sections we define the result metrics that are used, and present the results of each metric and compare it with other works. The results show the effectiveness of using sensor network WS in smart grid applications in terms of efficiency and energy saving.

5.2 Metrics Measures

Since we are talking about energy saving applications, we will have two types of metrics: the first one is the system performance which is related to the applications time efficiency, in addition to code and message sizes. The second category is the energy savings, which shows the amount of the energy saving produced by the energy saving applications. In general the result metrics will be the following:

- 1 : Sensor Node code sizes.
- 2 : Message Sizes.
- 3 : Completion Time.
- 4 : Energy Saving metrics.

5.3 Sensor Node code sizes

The sensor node hosts the web service and communicate with the web server. In addition, the memory sizes and the processor capability of the sensor node are limited, therefore, the code size on the sensor node has to be minimized, in addition to do as few process in the sensor side as possible.

Compacting name[1] strategy, is done by compacting the method names and arguments on the sensor node. In that way, the code sizes and the messages sizes decrease dramatically. In the WS, the method names and arguments affect not only the code size, but also the message size, because the WS stack produce the WSDL file and the SOAP message with respect to WS methods. So, the small code size will generate a light message.

The WS uses SOAP message to communicate between the requester and the provider. In the applications, the SOAP messages are used to communicate and transfer the data

between the web server and the sensor node, the following subsections will show the the SOAP messages in different applications.

5.3.1 Reading Meter Remotely Messages

In this application, the web server generates the following SOAP request message to ask about the current meter reading:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" >
<S:Header/>
<S:Body>
<ns2:operation xmlns:ns2="http://sensor/" />
</S:Body>
</S:Envelope>
```

Once the sensor receives the request message, it sends a SOAP message back to the server containing the meter reading as the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" >
<S:Body>
<ns2:operationResponse xmlns:ns2="http://sensor/" >
<return>40</return>
</ns2:operationResponse>
</S:Body>
</S:Envelope>
```

The above request and response SOAP messages, show that the sizes are compatible with the sensor node constraints. Since the Request message size is 218 KB and the response message size is 275. In addition, the response message contains the current

meter reading which is 40 kWh.

Therefore, Reading Meter Remotely application, needs two messages the first one is the request meter message the web server sends to the sensor node, and the second responds message the sensor node sends to the web server. Hence, this application needs at most $2*n$ messages, where n is the number of the houses in the application.

5.3.2 Remote Appliances Control Messages

The Remote Appliances Control application, needs the web server to communicate with the home sensor node combined to the AC to get the current reading, in order to do so, the Web server send the following SOAP request to the sensor node:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" >
<S:Header/>
<S:Body>
<ns2:AC1Rdng xmlns:ns2="http://sensor/" />
</S:Body>
</S:Envelope>
```

After the sensor node receives the requests from the server, it responds back with a SOAP response containing the current AC reading as the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" >
<S:Body>
<ns2:AC1RdngResponse xmlns:ns2="http://sensor/" >
<return>10</return>
</ns2:AC1RdngResponse>
</S:Body>
</S:Envelope>
```

The request and response messages sizes above show that they have the sizes of 275,275 KB respectively. On the other hand, if the web server asks the AC to decrease its consumption down, then, two other messages will be generated. The first one is the request message size from the server to the sensor, which has the new value of the consumption. The message is the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" >
<S:Header/>
<S:Body>
<ns2:SetAC1 xmlns:ns2="http://sensor/">
<value>5</value>
</ns2:SetAC1>
</S:Body>
</S:Envelope>
```

Once the sensor node receives the server request message, it responds back by the following message to show that the process was completed successfully:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" >
<S:Body>
<ns2:SetAC1Response xmlns:ns2="http://sensor/">
<return>5</return>
</ns2:SetAC1Response>
</S:Body>
</S:Envelope>
```

5.3.3 Managing Renewable Energy Completion Time

When the application starts, the web server sends the following request SOAP message to the sensor node to get the storage amount:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" >
<S:Header/>
<S:Body>
<ns2:MngRen xmlns:ns2="http://sensor/" />
</S:Body>
</S:Envelope>
```

The WS located on the sensor node sends the response back to the server contains the storage energy amount:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" >
<S:Body>
<ns2:MngRenResponse xmlns:ns2="http://sensor/" >
<return>13</return>
</ns2:MngRenResponse>
</S:Body>
</S:Envelope>
```

The sizes of the request and response SOAP messages are: 215, 269 KB respectively. In addition, the response message shows the amount of energy in the storage device which is in this case 13 kWh.

5.4 Sensor Node WSDL Files

The WSDL is the WS description files which contains the necessary information about the WS, such information is important for the client in order to access the WS, like the method name and the server address. In the following, we will show the WSDL files for each application and how it fits with the sensor node capabilities, in addition to its description.

5.4.1 WSDL for Reading Meter Remotely

The following XML, represents the WSDL file for the Remote Meter Reading WS, The file contains the method names *GtMetRdng*, which the web server need to access the sensor WS. In addition, The file contains the host address which is represented in the tag *soap:address location*. The file size is 1,909 bytes, and this size is suitable to the Pacemate sensor node[2] .

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--
Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2-hudson-752-.
-->
Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2-hudson-752-.
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://sensor/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://sensor/" name="GtMetRdngService">
<types>
```

```

<xsd:schema>
  <xsd:import namespace="http://sensor/"
schemaLocation="http://localhost:8080/SmarGridApps_Sensor/GtMetRdngService?xsd=1"
/>
</xsd:schema>
</types>
<message name="operation" >
  <part name="parameters" element="tns:operation" />
</message>
<message name="operationResponse" >
  <part name="parameters" element="tns:operationResponse" />
</message>
<portType name="GtMetRdng">
<operation name="operation" >
  <input wsam:Action="http://sensor/GtMetRdng/operationRequest"
message="tns:operation" />
  <output wsam:Action="http://sensor/GtMetRdng/operationResponse"
message="tns:operationResponse" />
  </operation>
</portType>
<binding name="GtMetRdngPortBinding" type="tns:GtMetRdng">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
<operation name="operation" >
  <soap:operation soapAction="" />
<input>
  <soap:body use="literal" />
</input>
<output>
  <soap:body use="literal" />
</output>
</operation>
</binding>
<service name="GtMetRdngService">
<port name="GtMetRdngPort" binding="tns:GtMetRdngPortBinding">

```

```

    <soap:address location="http://localhost:8080/SmarGridApps_Sensor/GtMetRdngService"
/>
  </port>
</service>
</definitions>

```

5.4.2 WSDL for Remote Appliances Control

The WSDL representation contains the methods names. Since we are implementing this application to control three homes AC's, three *ACRdng* methods will be implemented, one for each home. In addition the method *SetAC* will be implemented for each home. The number of messages for this application, if the total AC consumption is not above the limit, then no action will be done and the web server do not have to communicate with the sensor WS again. In this case, the web server needs two messages for each home to communicate, one for the AC consumption request and another for the AC response that contains the AC consumption.

On the other hand, if the Web server has to decrease the AC consumption, it needs two additional messages, one to ask the AC to decrease the consumption, and another one from the sensor to the web server telling that the process was done successfully. So at most, Remote Appliances Control application will use four messages to work.

In general the number of messages used by the Remote Appliances Control is at most $4*n$ where n is the number of sensor node combined with the AC.

The WSDL file size of this application is about 2400 Bytes.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <!--
Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2-hudson-752-.
- <!--
Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2-hudson-752-.
- <definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-

```

utility-1.0.xsd

```

xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://sensor/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://sensor/"
name="GtACRdngService">
  <wsp:Policy wsu:Id="GtACRdngPortBindingPolicy" />
- <types>
- <xsd:schema>
  <xsd:import namespace="http://sensor/"
schemaLocation="http://localhost:8080/SmarGridApps_Sensor/GtACRdngService?xsd=1" />
  </xsd:schema>
  </types>
- <message name="AC1Rdng">
  <part name="parameters" element="tns:AC1Rdng" />
  </message>
- <message name="AC1RdngResponse">
  <part name="parameters"
element="tns:AC1RdngResponse" />
  </message>
- <message name="AC2Rdng">
  <part name="parameters" element="tns:AC2Rdng" />
  </message>
- <message name="AC2RdngResponse">
  <part name="parameters"
element="tns:AC2RdngResponse" />
  </message>
- <message name="AC3Rdng">
  <part name="parameters" element="tns:AC3Rdng" />
  </message>
- <message name="AC3RdngResponse">

```

```

    <part name="parameters"
element="tns:AC3RdngResponse" />
  </message>
- <message name="SetAC1">
  <part name="parameters" element="tns:SetAC1" />
  </message>
- <message name="SetAC1Response">
  <part name="parameters" element="tns:SetAC1Response" />
  </message>
- <message name="SetAC2">
  <part name="parameters" element="tns:SetAC2" />
  </message>
- <message name="SetAC2Response">
  <part name="parameters" element="tns:SetAC2Response" />
  </message>
- <message name="SetAC3">
  <part name="parameters" element="tns:SetAC3" />
  </message>
- <message name="SetAC3Response">
  <part name="parameters" element="tns:SetAC3Response" />
  </message>
- <portType name="GtACRdng">
- <operation name="AC1Rdng">
  <input wsam:Action="http://sensor/GtACRdng/AC1RdngRequest"
message="tns:AC1Rdng" />
  <output wsam:Action="http://sensor/GtACRdng/AC1RdngResponse"
message="tns:AC1RdngResponse" />
  </operation>
- <operation name="AC2Rdng">
  <input wsam:Action="http://sensor/GtACRdng/AC2RdngRequest" message="tns:AC2Rdng"
/>
  <output wsam:Action="http://sensor/GtACRdng/AC2RdngResponse"
message="tns:AC2RdngResponse" />
  </operation>

```

```

- <operation name="AC3Rdng">
  <input wsam:Action="http://sensor/GtACRdng/AC3RdngRequest" message="tns:AC3Rdng"
/>
  <output wsam:Action="http://sensor/GtACRdng/AC3RdngResponse"
message="tns:AC3RdngResponse" />
  </operation>
- <operation name="SetAC1">
  <input wsam:Action="http://sensor/GtACRdng/SetAC1Request" message="tns:SetAC1"
/>
  <output wsam:Action="http://sensor/GtACRdng/SetAC1Response"
message="tns:SetAC1Response" />
  </operation>
- <operation name="SetAC2">
  <input wsam:Action="http://sensor/GtACRdng/SetAC2Request" message="tns:SetAC2"
/>
  <output wsam:Action="http://sensor/GtACRdng/SetAC2Response" message="tns:SetAC2Response"
/>
  </operation>
- <operation name="SetAC3">
  <input wsam:Action="http://sensor/GtACRdng/SetAC3Request" message="tns:SetAC3"
/>
  <output wsam:Action="http://sensor/GtACRdng/SetAC3Response" message="tns:SetAC3Response"
/>
  </operation>
</portType>
- <binding name="GtACRdngPortBinding" type="tns:GtACRdng">
  <wsp:PolicyReference URI="#GtACRdngPortBindingPolicy" />
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
- <operation name="AC1Rdng">
  <soap:operation soapAction="" />
- <input>
  <soap:body use="literal" />
  </input>
- <output>

```

```
<soap:body use="literal" />
</output>
</operation>
- <operation name="AC2Rdng">
  <soap:operation soapAction="" />
- <input>
  <soap:body use="literal" />
</input>
- <output>
  <soap:body use="literal" />
</output>
</operation>
- <operation name="AC3Rdng">
  <soap:operation soapAction="" />
- <input>
  <soap:body use="literal" />
</input>
- <output>
  <soap:body use="literal" />
</output>
</operation>
- <operation name="SetAC1">
  <soap:operation soapAction="" />
- <input>
  <soap:body use="literal" />
</input>
- <output>
  <soap:body use="literal" />
</output>
</operation>
- <operation name="SetAC2">
  <soap:operation soapAction="" />
- <input>
  <soap:body use="literal" />
```

```

    </input>
- <output>
    <soap:body use="literal" />
  </output>
</operation>
- <operation name="SetAC3">
    <soap:operation soapAction="" />
- <input>
    <soap:body use="literal" />
  </input>
- <output>
    <soap:body use="literal" />
  </output>
</operation>
</binding>
- <service name="GtACRdngService">
- <port name="GtACRdngPort" binding="tns:GtACRdngPortBinding">
    <soap:address location="http://localhost:8080/SmarGridApps_Sensor/GtACRdngService"
  />
  </port>
</service>
</definitions>

```

5.4.3 WSDL for Managing Renewable Energy

The following WSDL file represents the WS description for Managing Renewable Energy application. The method names are indicated by the tag name *operation name*, where the location address is *address location*. The WSDL file size for this application is 1901 KB.

```

<?xml version="1.0" encoding="UTF-8" ?>
<!--
Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2-hudson-752-.
<!--

```

Generated by JAX-WS RI at <http://jax-ws.dev.java.net>. RI's version is JAX-WS RI 2.2-hudson-752-.

```
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://sensor/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://sensor/"
name="MngRenEnergService">
<types>
<xsd:schema>
<xsd:import namespace="http://sensor/"
schemaLocation="http://localhost:8080/SmarGridApps_Sensor/MngRenEnergService?xsd=1"
/>
</xsd:schema>
</types>
<message name="MngRen">
<part name="parameters" element="tns:MngRen" />
</message>
<message name="MngRenResponse">
<part name="parameters" element="tns:MngRenResponse" />
</message>
<portType name="MngRenEnerg">
<operation name="MngRen">
<input wsam:Action="http://sensor/MngRenEnerg/MngRenRequest" message="tns:MngRen"
/>
<output wsam:Action="http://sensor/MngRenEnerg/MngRenResponse"
message="tns:MngRenResponse" />
</operation>
</portType>
<binding name="MngRenEnergPortBinding" type="tns:MngRenEnerg">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
<operation name="MngRen">
```

```

    <soap:operation soapAction="" />
  <input>
    <soap:body use="literal" />
  </input>
<output>
  <soap:body use="literal" />
</output>
</operation>
</binding>
<service name="MngRenEnergService">
  <port name="MngRenEnergPort" binding="tns:MngRenEnergPortBinding">
    <soap:address
location="http://localhost:8080/SmarGridApps_Sensor/MngRenEnergService" />
  </port>
</service>
</definitions>

```

Table 5.1, illustrates the messages and the code sizes in the sensor node after compacting. The sizes shown are combatable with the sensor node constrains as the Pacemante sensor [2].

Table 5.1: The code & messages sizes(Bytes)

Method name	code size	Req. message size	Res. message size	WSDL file size
GtMetRdng	103	218	275	1909
ACRdng	87	275	270	2400
SetAC	124	267	268	2400
MngRen	497	215	269	1901

5.5 Completion Time

The completion time of the application is the time needed for a single process to be completed. It is calculated as follows: It starts when the web server generates a request

until it receives the response back from the sensor. Figure 5.1, illustrates the way that the completion time was calculated. It starts when the Web server generates the request message on $T=t0$ until the responds arrive back from the sensor on the time $T=t1$. Then, the completion time is the result of the subtraction of the arrival of the respond message from the sensor $T=t1$, from the initial time $T=t0$.

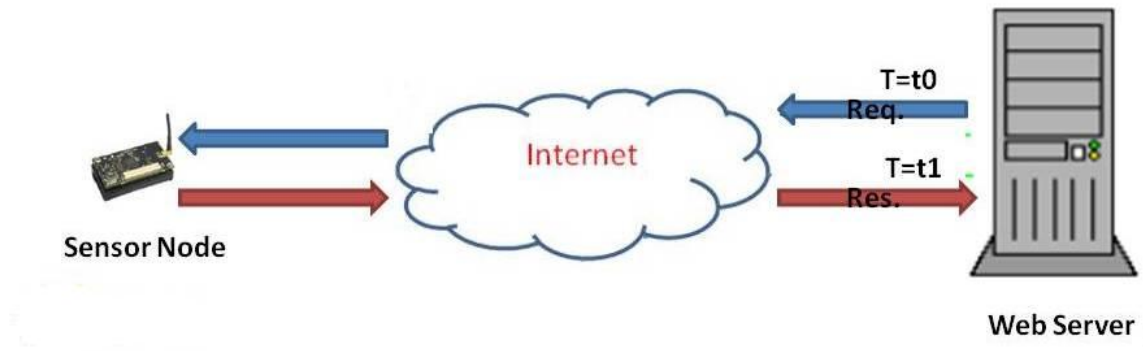


Figure 5.1: Completion time for A single process in Sensor WS application

Figures 5.2,5.3,5.4, represent the completion time for applications Remote Meter Reading, Remote Appliances Control, and Managing Renewable Energy respectively. The applications were run for 300 times and the completion time was registered. The three figures show that the completion time happens within the fractions of second. Furthermore, the completion time for the three applications are close to each other, the reason for similar completion times is due to the design approach, where most of the operations are done on the server side and the process that needs to be done on the sensor side WS takes relatively less time.

Table 5.2 summarize the completion time in milliseconds for the different applications, it gives the minimum, maximum, average, and standard deviation for the completion time values distribution. The reason for zero completion time appearing as the minimum in the application is due to test the system on a single machine, so a quite small time is needed for certain processes to be completed.

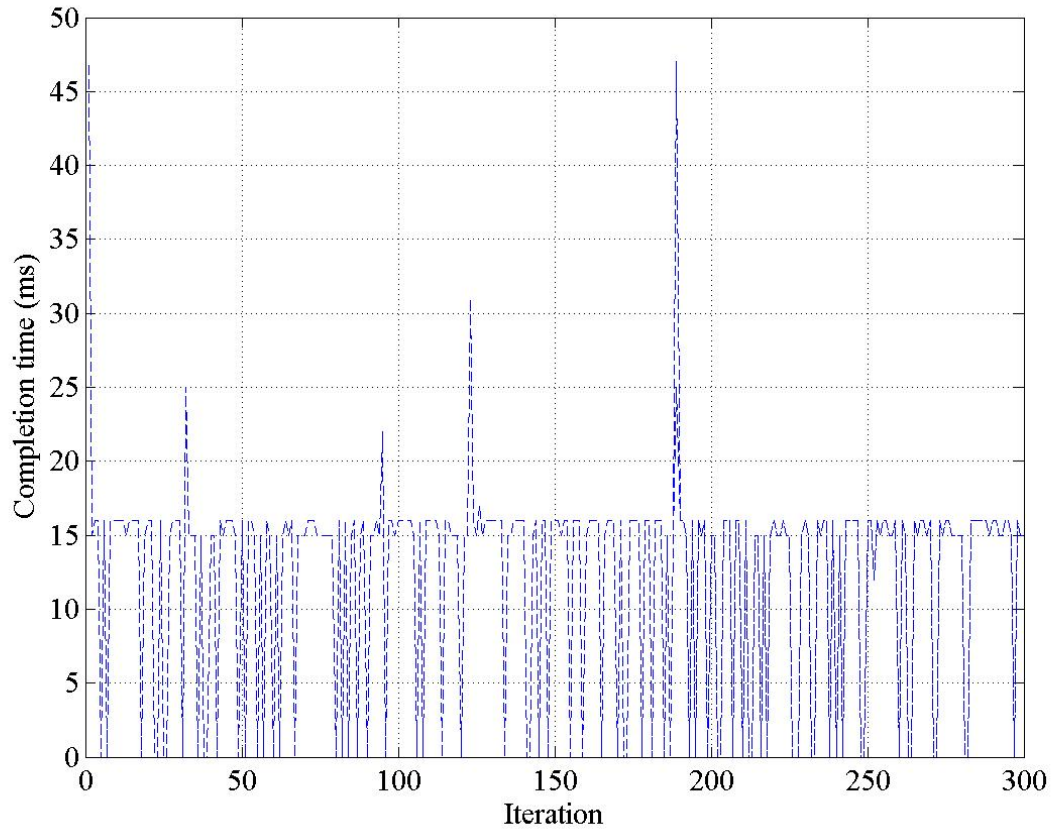


Figure 5.2: Completion time for the Reading Meter Remotely application

Table 5.2: Summary of the Completion values distribution

	Min	Max	Avg	Stdv
1st Application	0	47	12.67	7.04
2nd Application	0	31	11.51	7.02
3rd Application	0	47	11	8

5.6 Energy Saving

One of the main aspect of the smart grid applications is to reduce the electricity consumption and to save energy. The Remote Appliances control and Managing Renewable

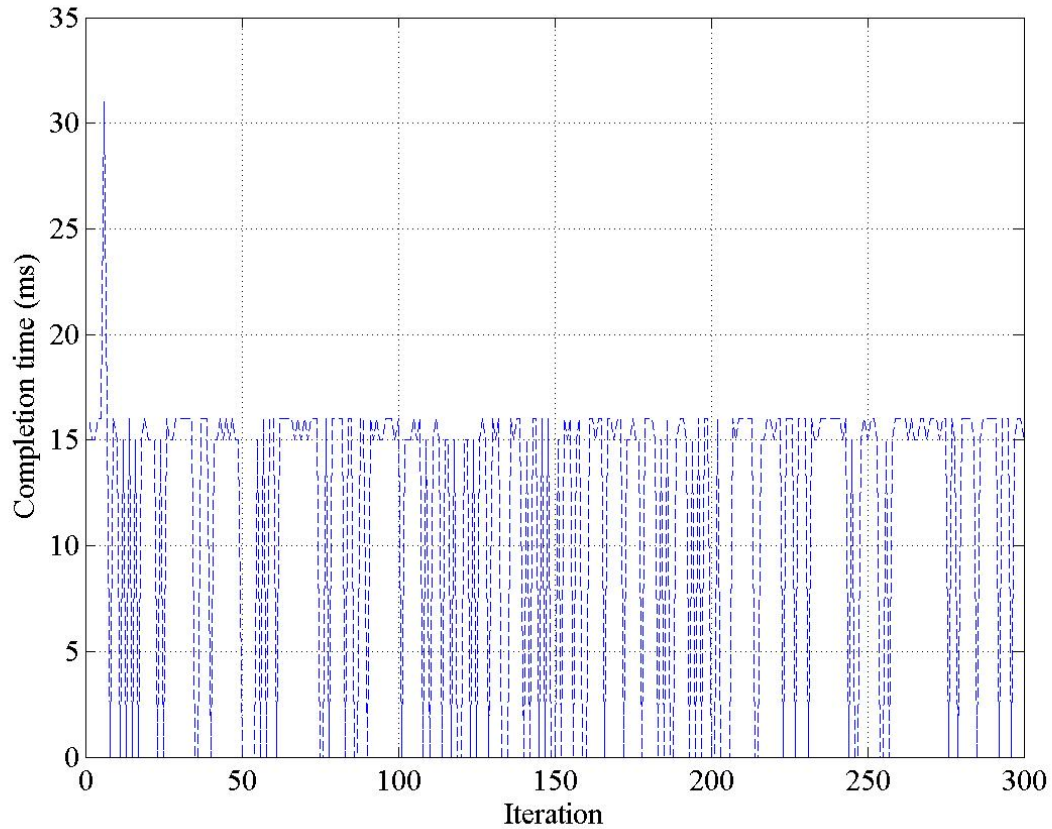


Figure 5.3: Completion time for the Remote Appliances Control application

Energy aim mainly to reduce the home energy consumption and to benefit from the Renewable energy resources. The following two subsections show the energy saving that could achieved by running the applications.

5.6.1 Remote Appliances Control Energy Saving

Figure 5.5, presents the amount of the energy saved by running the application for remote appliance control. We consider this application runs in three different smart houses. In each home, there is a sensor node combined with the smart AC, the sensor node runs a WS and is able to read the current AC consumption. This application is run once in a

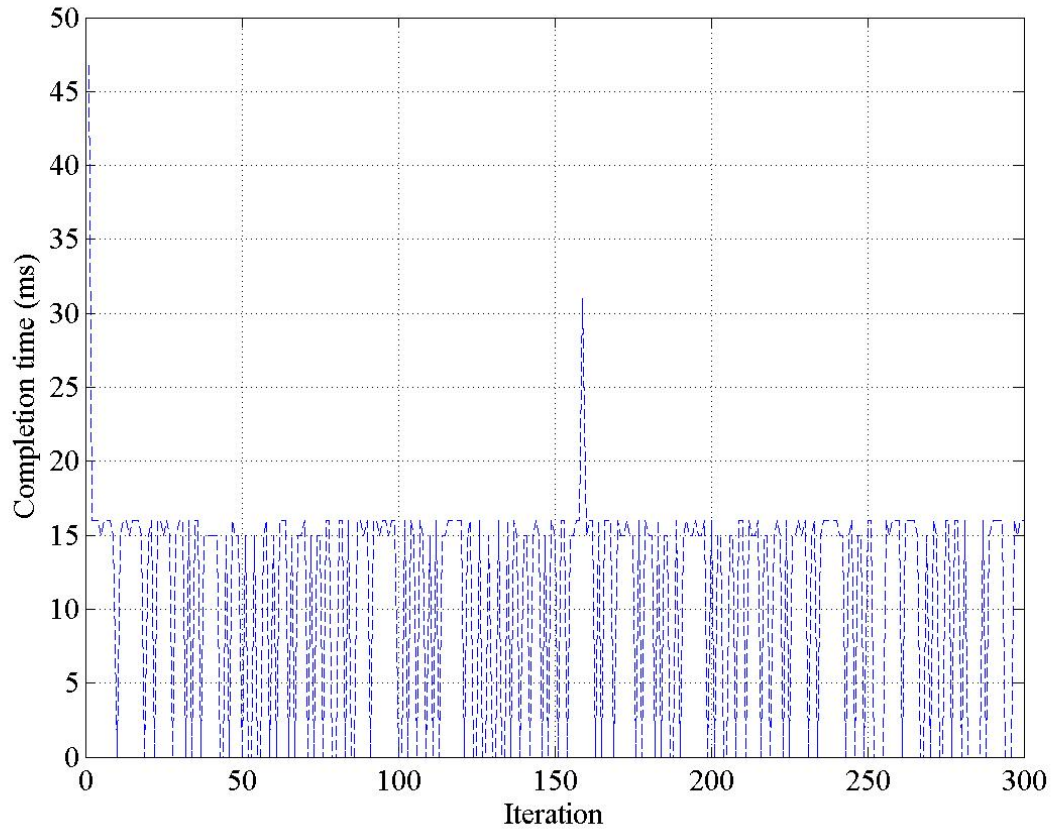


Figure 5.4: Completion time for Managing Renewable Energy application

day. We then show the amount of energy saved in 30 days. Note that the savings may be the same for several days because the application is designed to ask for a decrease in AC consumption only when the smart home is consuming more power than the average. Otherwise, if its consumption is equal or below the average the AC continues to run with user defined parameters. Fig.5.5 shows the energy saving in kWh by running the system in three different homes, each with an AC unit. Five periods were considered as peak hours, the first period lasts for three hours, the second one for five hours, the third for seven hours, the fourth for nine hours and the last one for eleven hours. For each peak period there is a maximum and a minimum AC consumption value. The central AC

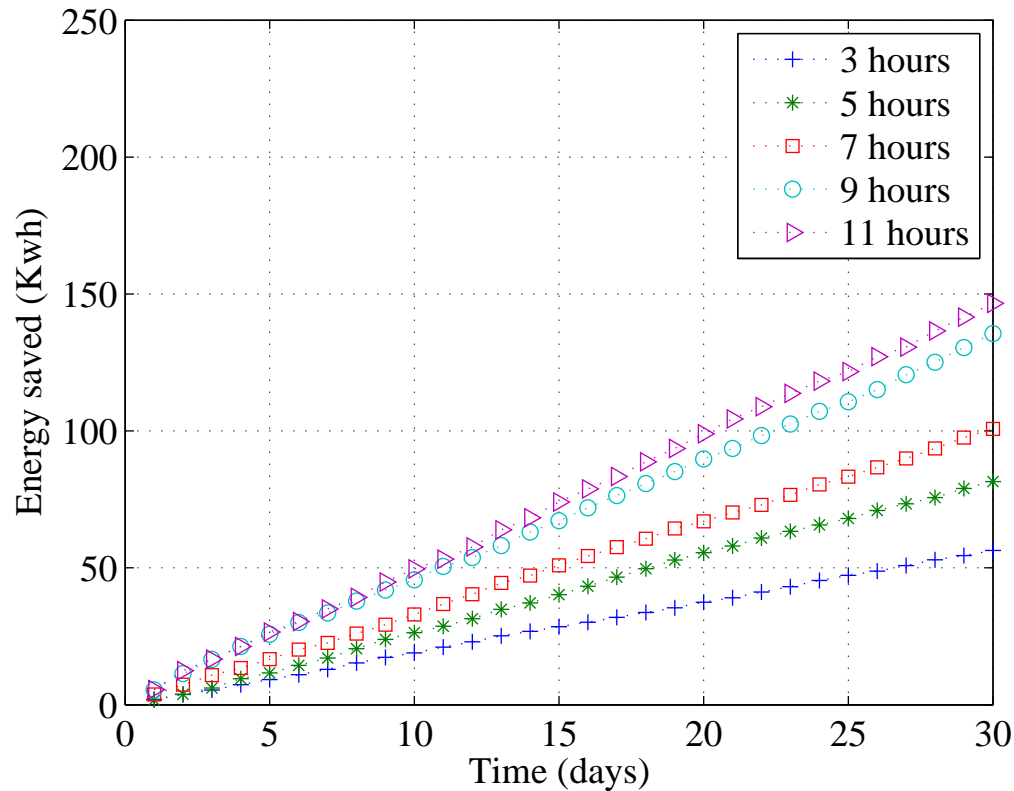


Figure 5.5: The amount of energy saving by running the Remote Appliances Control application.

wattage is approximately 3500W and the room AC is approximately 1050W[52]. After applying the AC wattage to the first peak period which lasts for three hours, we get the minimum value to be 3kWh and maximum value to 10kWh. In which case, the average is determined at a consumption value of 6.5 kWh, this is taken to measure the energy saving. However the figure shows the energy saving by running the application for 30 days for 100 times and taking the average.

5.6.2 Managing Renewable Energy Savings

Figure 5.6 presents the effect of managing renewable energy application in a smart home. The application is running in the peak hours and it takes the decision as to whether

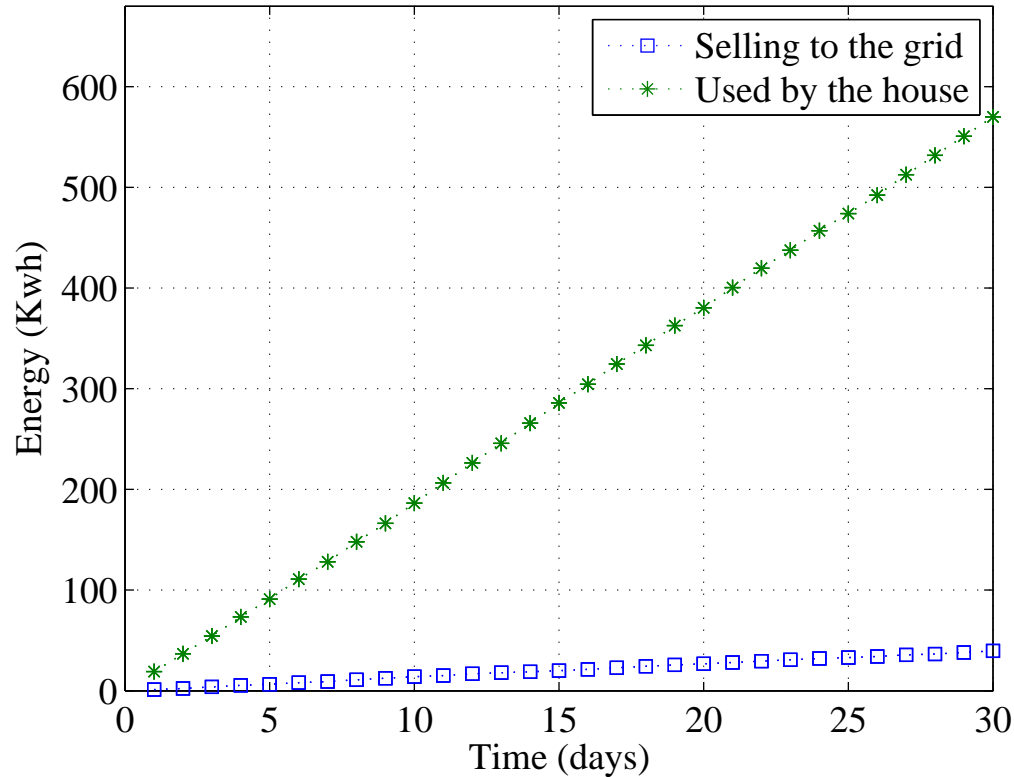


Figure 5.6: The amount of energy generating from the Renewable resources results from running Managing Smart Homes Renewable Energy

the excess electricity ought to be sold or stored, depending on given values determined before. The residential house consumes an average of 30kWh of daily according to the US Energy Information Administration[53]. Considering a sunny day where solar radiation is effective for 5 hours, and assuming a solar generator with a capacity between 1-3 kW[54], average solar production can be assumed to be 10kWh daily. In addition to that, a wind turbine is installed in the house. The amount of energy produced by a wind turbine can be measured depending on the following formula:

$$AEO = 0.01328 D^2 V^3 \quad [53],$$

Where AEO is the annual energy output kWh/year, D is the rotor diameter in feet, V is the annual average wind speed Mph. An average wind turbine has a diameter of

36 Ft. However the wind speed is different from one region to another, we assume an average wind speed of 8.6 Mph. After applying the above formula, this wind turbine generates approximately 10947.08 kWh per year, therefore the home can benefit from 30kWh daily wind energy.

By adding the two generated energy amounts, the house can benefit from a maximum of 40kWh of the renewable energy. The application runs for 30 days for a 100 times and the average is taken. The energy saving increases since it accumulates. It can be seen that using our application to manage the usage of the renewable energy, achieves a high saving, since at the end of the month there are about 600kWh of energy saving. This means almost two thirds of the energy consumed at home is supplied by the renewable energy savings. Furthermore the householder hydro bill would decrease because of selling back about 50kWh at the end of the month.

5.7 Summary

This chapter presented the system efficiency and the saving energy for the applications: Reading Meter Remotely, Remote Appliances Control, and Managing Renewable Energy.

The sensor node code and message sizes were presented and discussed. The sizes of the SOAP messages generated by the WS, the WSDL files, and the SOAP messages volume were tested. The sizes results showed that the sizes are compatible with the sensor node characteristics.

Another metric was the completion time, it measures the system efficiency in term of the time the process needed to be done. We measured the completion time in millisecond for the three applications, by running each application for 300 times and registered the completion time.

The last metric presented, was the energy saving. We measured the energy saving for Remote appliances control application, by running the applications for 30 days in three

different homes and then registering the energy saving amount, and thus showing that savings introduced by our applications are more for regions or seasons that have longer peak hours. In addition, we presented the energy saving yields from managing renewable energy application, showing that our renewable energy application increases the savings of the users.

Chapter 6

Conclusions

6.1 Summary and Concluding Remarks

The main objective of this thesis is to present smart grid applications using sensor network WS, and to measure the energy saving and evaluate the system efficiency yields from implementing those applications. The three applications that were introduced are:

- 1 : Reading Smart Meter Remotely
- 2 : Remote Appliances Control
- 3 : Managing Smart Homes Renewable Energy

Implemented WS in WSN present many challenges: the sensor memory, the processor, and the power constraints. To solve these challenges many techniques were introduced, including compression of the XML file to fit the sensor node, and decreasing the code and message sizes on the sensor side. On the other hand, sensor WS has many advantages, since it is platform independent, easy to develop and easy to integrate with other web applications.

The smart grid is the integration of the advanced technology with the existing power grid to introduce a communication between the different points on the grid. It aims

to decrease the energy consumption, and to increase the transparency and reliability. Demand response where the utility generators will work properly during high demand periods, is another aspect of the smart grid. Furthermore, the end user will be an active element in the process of power consumption and generation.

The system model proposed on our work, consists from several smart homes connected with each other. Every smart home deploys WSN. The sensor node is combined with the appliances. ZigBee protocol is used as a communication protocol between the sensor nodes. The smart home is connected with the utility server through the gateway sensor node. On the other hand, the gateway node communicates with the home WSN via the WS.

The systems were analyzed and described. The first application, Reading Smart Meter Remotely, aims to make an automatic system that can take the meter reading periodically and produce a monthly bill for the customer. On the other hand, the system enables the utility to perform Interval measurements to evaluate the home consumption over the day time. This system is integrated with the other utility systems like customer care and billing systems.

The second application that was proposed is Remote Appliances Control, the application allows the utility to arrange its generators during high demand periods. The idea of the application is the following: At the peak hours, the utility servers could control the AC consumption at the houses, if it goes beyond the allowed limit, based on a previous agreement with the inhabitant. The application starts reading the current AC consumption of the homes, this done by accessing the WS located on the sensor node combined with the AC. After finishing all the specified houses, the web server takes the reading average, if the average is less than the limit, then nothing happens, and the AC's continue to work normally. Otherwise, the application sent another message to all the AC's which exceed the consumption limit through the sensor node WS, asking the sensor to lower the AC electricity consumption.

The third application is Managing Renewable Energy. As the smart home benefits from the Renewable energy resources, like wind turbine and solar panel, this application manages the usage of the renewable energy. The application communicate with the sensor node combined with the storage device via the WS, after getting the current amount of energy on the storage it takes one of the following actions: The home keeps using electricity from the grid, or the home quits using electricity from the grid and uses locally generated energy. otherwise the home sells electricity to the grid.

The implementation is described in this thesis through a detailed explanation. First: The Sequence diagrams shows the message exchanged by the system actors, Second: The activity diagram shows the different events on the system. Third: the algorithm shows the ordered task which was introduced.

The system was implemented using java NetBeans and MySql database. The performance and the energy saving of the System were evaluated. To evaluate the code compatibility with the sensor node, we measure the code, SOAP messages, and WSDL file sizes for the different applications. In addition, the completion time measures the application efficiency by giving the time needed for the message to travel from the web server to the sensor node and come back again. The last metric was the amount of energy saving results from running Remote Appliances Control, and Managing Renewable energy, we showed that the energy saving introduced by the mentioned applications increased when peak hours increased. Thus, both applications enhance the savings for the user.

6.2 Future Research

We implemented this applications using SOAP WS technology. However, several ideas can be promising for future research in the field of Smart Grid applications using sensor WS. The current research can be extended to cover the following topics:

- 1 : implement the RESTfull protocol and analyze its performance in sensor network

WS, and measure its efficiency on the Smart grid applications.

2 : Include other smart grid applications: especially Plug-in Hybrid Electrical Vehicles (PHEV), by testing the appropriate time of charging the car and the suitable place for charging, and measure the energy saving and the system efficiency.

3 : Study and investigate the security issues on Sensor network WS, and apply the security techniques to the messages. In addition to set a certain authentication rule on the WS to prevent any harmful access.

Bibliography

- [1] N. Priyantha, A. Kansal, M.Goraczko, and F.Zhao, “Tiny web service: Design and implementation of interoperable and evolvable sensor networks,” in *Proc. of the 6th ACM conference on Embedded network sensor systems*, (Raleigh, NC, USA), June 2008.
- [2] N. Glombitza, D. Pfisterer, and S.Fischerr, “Integrating wireless sensor networks into web service-based business processes,” in *Proc. of the 4th International Workshop on Middleware Tools, Services and Run-Time Support for Sensor Networks*, (Urbana Champaign, Illinois, USA), 2009.
- [3] J. Scott, P. Vaessen, and F. Verheij, “reflections on smart grids for the future.” <http://www.kema.com/nl/papers/Reflections.aspx>, 08.
- [4] F. Rahimi and A. Ipakchi, “Demand response as a market resource under the smart grid paradigm,” *IEEE TRANSACTIONS ON SMART GRID*, pp. 82–88, June 2010.
- [5] C.Warmer, K. Kok, S. Karnouskos, A. Weidlich, D. Nestle, P. Selzam, J. Ringelstein, A. Dimeas, and S. Drenkard, “Web services for integration of smart houses in the smart grid,” in *Proc. of Grid-Interop Conference*, (CO, USA), 2009.
- [6] “Electricity supply information for ontario hydro consumers,” 2010. <http://www.ontario-hydro.com/>,.

- [7] M.Hubbert, “Nuclear energy and the fossil fuels,” in *Presented before the Spring Meeting of the Southern District Division of Production, American Petroleum Institute*, (San Antonio, Texas, USA), 1956.
- [8] “Insitechinc.” http://www.insitechinc.com/what-is/WhatIsXTT_technologyPerspective.html/.
- [9] D. Leeds, “The smart grid in 2010: Market segments, applications and industry players.” <http://www.gtmresearch.com/>, 10.
- [10] W3C, “Ws definition.” <http://www.w3.org/TR/ws-arch/>, 2004.
- [11] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, “Web services description language (wsdl) 1.1.” <http://www.w3.org/TR/wsdl>, 2001.
- [12] P. Gibbons, B. Karp, Y. Ke, S. Nath, and S. Seshan, “Irisnet: An architecture for a worldwide sensor web,” *Pervasive Computing, IEEE*, no. 2 Issue:4, pp. 22–33, 2003.
- [13] K. Delin, S.Jackson, and R.Some, “Sensor webs,” *NASA Tech Briefs*,, p. 80, 1999.
- [14] K.Delin and S.Jackson, “The sensor web: A new instrument concept,” in *Proc. of SPIEs Symposium on Integrated Optics*, (San Jose, CA,USA), January 2001.
- [15] C.Werner, C.Buschmann, and S.Fischer, “Advanced data compression techniques for soap web services,” in *Modern Technologies in Web Services Research*, pp. 76–97, USA: Liang-Jie Zhang (IBM T.J. Watson Research), 2007.
- [16] D. Pfisterer, M. Wegner, H. Hellbruck, C. Werner, and S. Fischer, “Energy-optimized data serialization for heterogeneous wsns using middleware synthesis,” in *Proc. of The Sixth Annual Mediterranean Ad Hoc Networking Workshop*, (Corfu, Greece), pp. 180–187, June 2007.
- [17] “Sensor model language sml,” August 2010. <http://www.botts-inc.net/vast.html>.

- [18] C. Augeri, B. Mullins, L. Baird, D. Bulutoglu, and R. Baldwin, “An analysis of xml compression efficiency,” in *Proc. of ACM Workshop on Experimental Computer Science*, (San Diego, California, USA), pp. 180–187, June 2007.
- [19] S. Maneth, N. Mihaylov, and S. Sakr, “Xml tree structure compression,” in *Proc. of 19th International Conference on Database and Expert Systems Application, DEXA '08.*, (Turin), 2008.
- [20] A. Dunkels, “Full tcp/ip for 8-bit architectures,” in *Proc. of the 1st international conference on Mobile systems, applications and services*, (New York, USA), 2003.
- [21] A. Dunkels, J. Alonso, and T. Voigt, “Making tcp/ip viable for wireless sensor networks,” in *Work-in-Progress Session of the first European Workshop on Wireless Sensor Networks*, (Berlin, Germany), 2004.
- [22] M. Anagnostou and E. Protonotarios, “Performance analysis of the selective repeat arq protocol,” *IEEE Transactions on Communications*, no. Issue:2, pp. 127–135, 1986.
- [23] W3C, “Bpel orchestration of rule-based web services.” http://www.w3.org/2005/rules/wg/wiki/UCR/BPEL_Orchestration_of_Rule-Based_Web_Services.html, 00.
- [24] W3C, “Simple object access protocol soap.” <http://www.w3.org/TR/soap/>, 2007.
- [25] R. Thomas, *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.
- [26] Y. Dogan and D. Adam, “Efficient application integration in ip-based sensor networks,” in *Proc. of the first ACM Workshop On Embedded Sensing Sys. For Ene.-Efficiency In Buildings*, (CA, USA), 2009.

- [27] I.Amundson, M.Kushwaha, X.Koutsoukos, S.Neema, and J.Sztipanovits, “Efficient integration of web services in ambient-aware sensor net. applications,” in *Proc of the 3rd International Conference on Broadband Communications, Net. and Sys. (BROADNETS’06)*, (CA,USA), 2006.
- [28] G.Moritz, E.Zeeb, S.Prter, F.Golatowski, D.Timmermann, and R.Stoll, “Devices profile for web services and the rest,” in *Proc. of 8th IEEE International Conference on Industrial Informatics (INDIN)*, (Osaka, Japan), 2010.
- [29] M.Takahashi and T.Bin, “Litews: A web service enhancing multi-user queries in data intensive sensor networks,” in *Proc. of the IEEE International Conference on Communications (ICC’10)*, (Cape Town, South Africa), 2010.
- [30] T.Garrity, “Getting smart,” *IEEE Power and Energy magazine*, pp. 38–45, 2008.
- [31] “the now house project.” <http://www.nowhouseproject.com/>, 2008.
- [32] “ontario power authority, feed-in-tarrif program.” <http://fit.powerauthority.on.ca/>, 2006.
- [33] “Smart-a project.” <http://www.smart-a.org/>, 2006.
- [34] D.Infield, J. Short, C. Home, and L. Freris, “Potential for domestic dynamic demand-side management in the uk,” in *IEEE Power Eng. Soc. General Meeting*, (Tampa, FL , USA), pp. 1–6, 2007.
- [35] J.Paateron and P.Lund, “A model for generating household electricity load profiles,” *International Journal of Energy Research*, pp. 273–290, 2006.
- [36] M.Erol-Kantarci and H.Mouftah, “Using wireless sensor networks for energy-aware homes in smart grids,” in *Proc. of IEEE Symposium on Computers and Communications (ISCC’10)*, (Riccioni, Italy), 2010.

- [37] A.Bose, “Smart transmission grid applications and their supporting infrastructure,” *IEEE TRANSACTIONS ON SMART GRID*, pp. 11–19, June 2010.
- [38] “Distributed generation in liberalised electricity markets.” <http://www.iea.org/textbase/nppdf/free/2000/distributed2002.pdf>, 2002.
- [39] A.Molderink, V.Bakker, M.Bosman, J.Hurink, and G.Smit, “Management and control of domestic smart grid technology,” *IEEE TRANSACTIONS ON SMART GRID*, pp. 109–119, September 2010.
- [40] I.Akyildiz, W.Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks Elsevier Journal*, p. 393422, March 2002.
- [41] “Daintree networks: Introducing zigbee rf4ce. daintree networks.” <http://www.daintree.net/downloads/whitepapers/zigbee-rf4ce-intro.pdf>.
- [42] L.Pengfei, L.Jiakun, N.Luhua, and W. Bo, “Research and application of zigbee protocol stack,” in *Proc. of International Conference on Measuring Technology and Mechatronics Automation*, (CO, USA), 2010.
- [43] N.Dou, Y.Mei, Z.Yanjuan, and Z.Yan, “The networking technology within smart home system –zigbee technology,” in *Proc. of International Forum on Computer Science-Technology and Applications*, (Chongqing, China), 2009.
- [44] t. R. E. P. N. f. t. s. C. REN21, “Renewables global status report: 2009,” 2009. <http://www.ren21.net/>.
- [45] “Netbeans ide.” <http://netbeans.org/>, 2009.
- [46] “Mysql open source data base engine.” <http://www.mysql.com/>, 2009.
- [47] “Metro web services.” <http://metro.java.net/>.
- [48] “Java 2 platform, enterprise edition (j2ee) overview.” <http://java.sun.com/j2ee/overview.html/>.

- [49] “Jdbc.” <http://java.sun.com/products/jdbc/overview.html/>”.
- [50] “Mysql connectorj.” <http://dev.mysql.com/doc/refman/5.1/en/connector-j.html/>”.
- [51] T.Lethbridge and R.Laganriere, *Object-Oriented Software Engineering*. The McGraw Hill Company, 2001.
- [52] “Ministry of energy and infrastructure ontario.” <http://www.mei.gov.on.ca/>”.
- [53] “Us department of energy.” <http://www.energy.gov/>”.
- [54] “Go solar: A program of the clean air.” <http://www.gosolarontario.ca/>”.