Experimental analysis of an event tracking energy-efficient WSN

Ravi Kishore Kodali
Department of Electronics and Communications Engineering
National Institute of Technology, Warangal
Andhra Pradesh, 506004, India
E-mail: ravikkodali@gmail.com

***Abstract*—In Wireless sensor networks (WSN's), various tiny wireless sensor nodes or motes collect the data from the surrounding areas, where they are deployed and forward the same towards the gateway. Various pieces of information, such as occurrences of various events in the field and associated aggregate values need to be extracted from the data collected by these motes in an end-to-end WSN solution. Furthermore, making this information available to authenticated remote users with reduced latencies and errors, is a challenging task. In this work, a an experimental WSN set up using light and accelerometer sensors is given and the experimental results of the same are presented. This paper also provides an analysis of *XMesh* protocol used in the experimental set up. In order to make the data collected by the WSN, accessible to a remote user, a solution based on a portable computer board, *Raspberry Pi*, is also given.**

Keywords:WSN, mesh topology, multi-hop routing, event monitoring, accelerometer

I. INTRODUCTION

Wireless sensor network (WSN) technology strives to solve the problem of tracking and monitoring various events in remote and hazardous areas [1],[2],[3]. A WSN consists of resource constrained sensor nodes, which are used to collect the data from their surroundings and the nodes are deployed randomly in outdoor applications [4], [5], [6]. As the nodes are small and are deployed in remote locations, the computational and energy constraints of the nodes make the design of WSN protocols a difficult task. Various network protocols, used in computer networks, cannot be directly used in WSNs. Various routing and MAC layer protocols are proposed in the literature to meet the requirements of WSN applications [7], [8].

In WSNs, the data collected by sensor nodes is forwarded towards the gateway node. The gateway node connects the WSN to a computer network. The gateway node, in turn is connected a base station (BS). The BS is used to monitor and to control the data flow in the network and also carries the responsibility of forwarding the collected data to the remote users, making use of its connectivity to the Internet. Figure 1 presents various tiers in a WSN based end-to-end application with web connectivity. The mote tier comprises of sensor nodes deployed in the field. The nodes make use of IEEE 802.15.4 MAC protocol to communicate among themselves and with the gateway node. The gateway node is connected to a portable *Raspberry Pi* based computer board [9], which is powered by $5V/700$mA power supply. This board makes use of a Linux operating system and runs a lightweight web hosting software. This board also stores the data sent by the gateway node. The most recent data is made available to remote users making use of its web interface. The *Raspberry Pi* gains access to the wireless local area network (WLAN) with the help of 802.11n Wi-Fi module connected to its USB port. Further, it is connected to the Internet through local area network (LAN) and a proxy server. An authenticated remote user can access the data through the web interface of the *Raspberry-Pi* board. The data is also sent to the server tier, which consists of a data logger along with a data parsing and a naming server. The parsed data is forwarded to the client tier. In the client tier, a graphical user interface (GUI) for the data presentation and its analysis is available.

In this work, an event monitoring WSN application is considered. The WSN application consists of a static network comprising of static nodes and mobile sensor nodes. By making use of the dynamic multi-hop routing protocol, *XMesh* [10], various static nodes are connected to the gateway node. The application aims at collecting the data from a mobile node. The static nodes receive the data from a mobile node and relay the same towards the BS. Further, this data is presented to the remote user through the BS and the same is then analysed using *Moteview* software platform. The *Raspberry Pi* board facilitates in sending the data from the WSN to a remote user. With the help of an inexpensive *Raspberry Pi* board, the data in the WSN is made available on the Internet without a needing a separate computer system near the BS. This makes the concept of *Internet of things* (IoT) [11] easily realizable in an inexpensive manner. The rest of the paper is organized as follows: Section II gives an insight of the *XMesh* routing protocol used in the application. Section III discusses the hardware used in the experimental set up of WSN application. The experimental set up is given in section IV. An analysis of the experimental results is presented in section V. Section VI concludes the paper and a scope for future work is given.

II. XMESH PROTOCOL

XMesh is an ad-hoc, multi-hop networking protocol for Wireless sensor networks(WSNs) [12],[13]. It consists of

978-1-4673-6217-7/13/$31.00 ©2013 IEEE

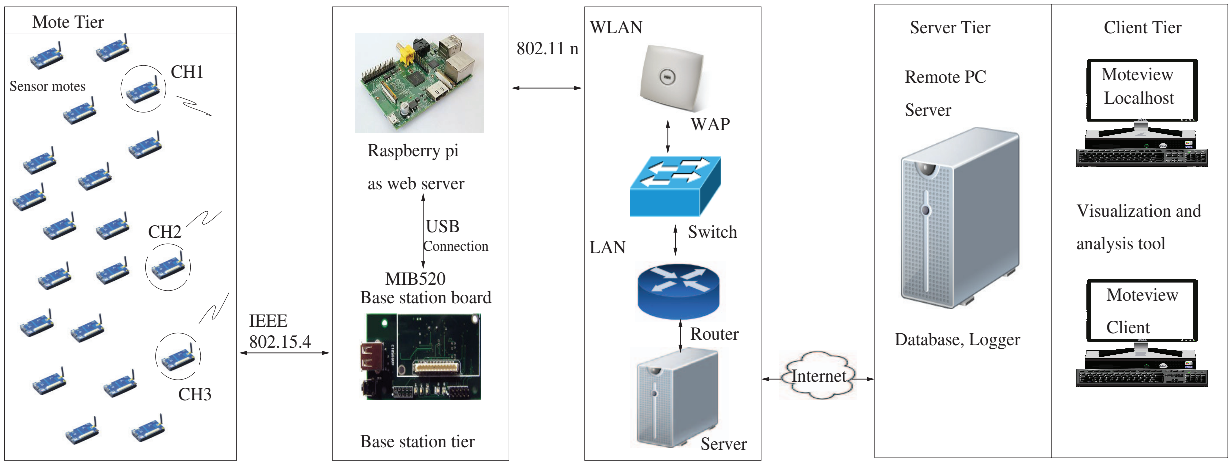

Fig. 1. Wireless sensor network scenario

many sensor motes and a base station. XMesh is self-heal able and self-sustainable protocol, which supports addition and deletion of nodes at any instance [14]. XMesh implements mesh topology [15], in which motes can route the data and efficiently establish end-to-end connection. To determine various routing paths, the cost metric (CM) [16] representing link quality, which is calculated by a receiving mote using equation (1).

$$CM_{link_{q}uality} = \frac{n_R}{n_E}, \tag{1}$$

where, n_R is the number of messages received and n_E is the number of expected messages. The number of expected messages is calculated by keeping track of the sequence number of the message from a neighbouring mote. By updating the link quality parameter periodically, route from a parent node to a sink node is selected avoiding bad links. The multi-hop routing [17] supported by the *XMesh* protocol not only guarantees end-to-end connectivity but also reduces the energy cost of a message transmission. The *RouteSelect* interface of the protocol is used to select an efficient route, while the *RouteControl* interface is used to monitor the route quality and modify the route state update interval.

Xmesh protocol can be implemented in three power modes, namely high power (HP), low power (LP) and extended low power (ELP). In both the HP and LP modes, all of the multi-hop mesh networking capabilities are enabled, while in ELP mode motes can not route the data and is connected to the gateway using single hop alone. The communication latency in LP mode is more because on board peripherals are kept in *SLEEP* mode for most of the time to conserve battery energy. Also, in low power mode indication the LEDs are turned off. The operation mode can be selected at the time of initializing the node. For example, to force low power routing on a *iris* motes, make command is

make iris route, lp

Xmesh protocol is implemented in the sensor motes and a gateway or a base station. The application programs are developed using *nesC* for different types of sensor motes. The base station makes use of the interfaces supported by the XMesh. The application code for sensor motes such as XMDA100, XMTS300, XMTS500 differ, as these have different sensor types. This work uses MDA100 board while carrying out various experiments and a brief analysis of the nesC code for the same is provided. In the application program of the MDA100, the initialization timer rate is set according to pre-defined sample rate. This initialization is done under atomic section. Various time critical operations are written under atomic section, which can not be interrupted and avoids instability caused due to overwriting of the global variables. Further, when the timer is fired, temperature, light and other sensor type data are measured. Various asynchronous events, such as, ADC data ready are implemented and respective *getData()* function is called, when an event occurs. While transmitting a message along with the data, node id, packet id and parent id are also transmitted. The *RouteControl* interface provides parent id. The *Send* interface selects the route and transmit the message. The broadcast messages are handled by *XCommand* interface and for different commands like *SETRATE, SLEEP,WAKEUP*, appropriate actions are taken. The application program for base station forwards the packet for any XMesh application and inject XCommand packets into the network. The following command is used to build base station application code on *iris* platform with 2.41 GHz frequency.

make iris base freq,2.410

III. WSN HARDWARE

In this work, IRIS sensor mote [18] with MTS400 sensor board is used. The MTS400 sensor board is installed on the mobile sensor node. For motion detection two-axis accelerometer is present on the sensor board. Also, to capture the surrounding environment humidity, temperature and light sensor are provided with the sensor board. Figure 2 shows the IRIS mote along with its MTS400 sensor board.

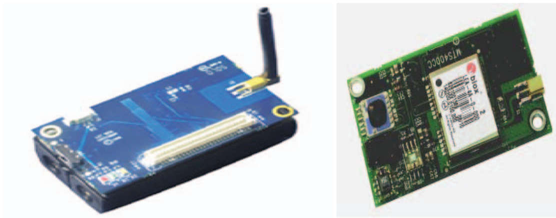

Fig. 2. Sensor board

Sensor type	Specification
Temperature	Accuracy: $\pm 2^o C$ Range: $-40^o C$ to $80^o C$
Light sensor	ON resistance (white light) $10K\Omega$ OFF Resistance $520K\Omega$
Pressure	Accuracy: ± 3.5 Range: 300 to 110 mbar
Accelerometer	Range: $\pm 2G\ (G = 9.81m/s^2)$ Sensitivity: 167 mV/G

TABLE I
CHARACTERISTICS OF THE SENSOR BOARD- MTS400

The IRIS mote has Atmel's AT1281 micro-controller with Atmel's AT86RF230 transceiver based on IEEE 802.15.4 protocol and Zigbee compliant radio [19]. The RF transceiver of the mote can be tuned to different frequency channels ranging from 2.405 GHz to 2.48 GHz, each separated by 5 MHz (16 channels). The communication range of the mote can be set by varying the node's transmission power in the range of 3 dBm to -17.2 dBm. IRIS mote provides 4 KB RAM and 128 KB flash memory which is sufficient for most of the WSN applications. Apart from the flash memory on micro-controller, IRIS mote also contains external 512 KB flash memory to support *On The Air Programming* (OTAP) [20]. The programming board, MIB 520, is used to program IRIS mote making use of a USB port in a computer. This board has an on-board ATMega16 controller to program the mote. Programming of an IRIS mote requires TinyOS platform to be installed on the computer to which the programming board is connected. The TinyOS image along with the application code is downloaded to the mote through the ATMega16 controller. To enable programming using USB ports FTDI USB Virtual COM port drivers need to be installed in the computer.

Parameter	Value
RAM	8 KB
ROM	128 KB
Clock Frequency	7.37 MHz
Supply voltage	2.7 to 3 V
TX Power	-17 to 3 dBm
TX current consumption	16 mA
RX current consumption	15 mA

TABLE II
IRIS MOTE SPECIFICATIONS

A. Calibration of accelerometer

In this work, the object under observation is tied with a mote having MTS400 sensor board. To track various movements of the object, an accelerometer is used. The two-axis accelerometer available on the sensor board needs to be calibrated to detect the object movement with minimum error. The accelerometer is calibrated with respect to acceleration due to gravity $\left(g = 9.82m/s^2\right)$. Towards this calibration, the X-axis of the accelerometer is directed towards the earth surface in a perpendicular manner and the reading of the accelerometer is noted down as $+1g$. The reading of accelerometer, when directed away from the surface of the earth is taken as $-1g$. The same procedure is followed for the Y-axis calibration as well. Further, the readings obtained by both the axes of the accelerometer (acc_x, acc_y) are converted into the magnitude and angular values for the purpose of analysis. The magnitude (M) and angular (ϕ) calculations are carried out using equations (2) and (3), respectively.

$$M = \sqrt{(acc_x)^2 + (acc_y)^2} \tag{2}$$

$$\phi = \tan^{-1}\left(\frac{acc_y}{acc_x}\right) \tag{3}$$

IV. EXPERIMENTAL SET UP

The *IRIS* sensor motes, installed with the XMesh protocol are deployed in the area under observation. The deployed nodes are not in the range of the gateway and hence multi-hop paths are to be established during the initial stage. When the topology gets stabilised, the network becomes ready for monitoring the events in the range of the deployed network. The object to be tracked is attached with the sensor node, which is capable of measuring two-axis acceleration, temperature, humidity and light conditions of the surroundings. If the node is in the radio range of the gateway device, it will send its data directly to it. Otherwise, the mobile node attempts to establish a path with the gateway through one of the nearby static nodes deployed in the network. The mobile node moves along the deployed network and broadcasts its sensed data periodically. This data is collected by one of the nearby nodes of the network the same is forwarded towards the gateway. This data is analysed by making use of the *Moteview* software platform. Also, the data is sent to the *Raspberry Pi* board, which makes the data available to the authenticated users through the Internet.

V. ANALYSIS OF EXPERIMENTAL RESULTS

Moteview software is used to analyse the data collected by sensor motes. In this section, an experimental analysis carried out using WSN hardware is presented.

A. Multi-hop routing

The XMesh protocol supports multi-hop routing to conserve the energy of nodes and to provide guaranteed connectivity between sensor motes and the gateway. To test multi-hop routing capability of XMesh protocol, first all the nodes deployed nearer to gateway. Then, the node 4 is moved away

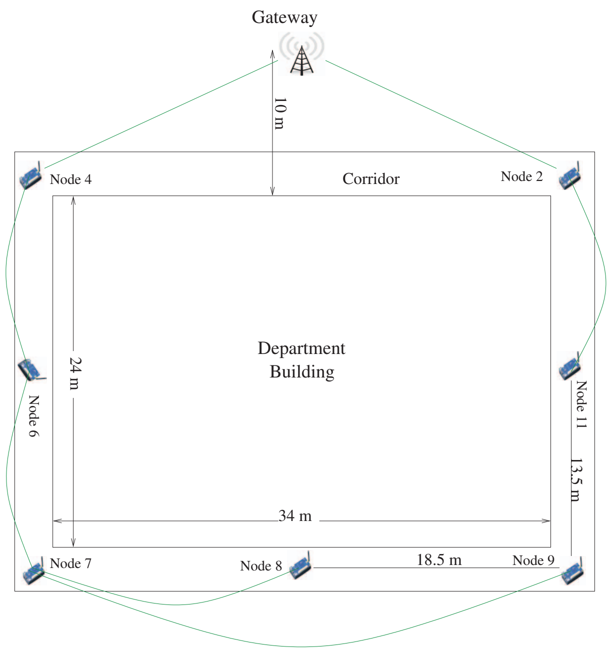

Fig. 3. Wireless sensor network deployment

from the gateway ($> 30m$ in indoor conditions). Figure 4 shows the multi-hop connectivity. Now, node 3 is the parent node for the node 4. If two nodes are kept long distance from

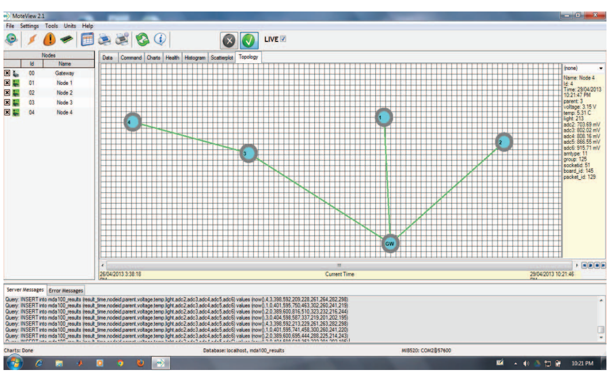

Fig. 4. Multi-hop routing in WSN (two hops)

the gateway $(node4 : 30m, node2 : 50m)$, a child node takes a total of three hops to get connected to the gateway. This experiment proves the reliability and the robustness of the XMesh protocol. But the time required to update the topology is on an average 1 minute, which may cause loss of data.

B. Node failure detection

The persistence of data collection from the mobile node depends on the robustness of routing protocol among the static sensor nodes. The response of XMesh routing protocol to the node failure is analysed by turning off the critical node in the topology. Figure 6 shows the network topology prior to turning off of the critical node.

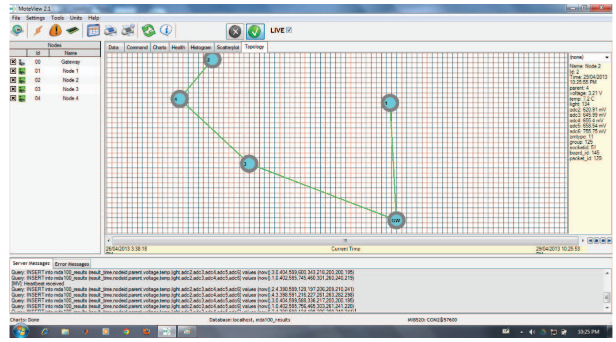

Fig. 5. Mulit-hop routing in WSN (3 hops)

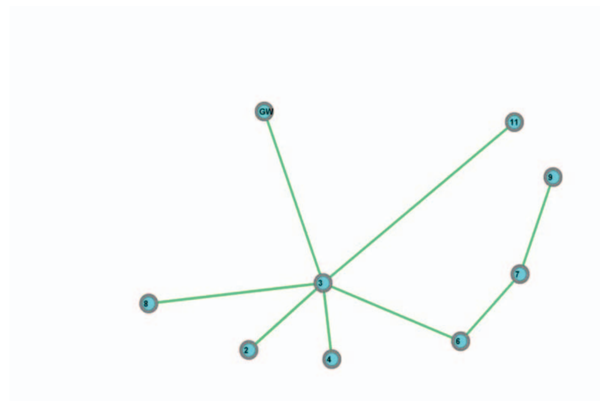

Fig. 6. Faulty node detection in Moteview

From figure 6, it is clear that the node 3 is the critical node in the network as all other nodes are connected to the gateway through it. The node 3 is turned off and the topological changes of the network are monitored. After 5 minutes, a stable topology has been arrived without any human intervention. The updated topology is shown in Figure 7. It proves that multi-hop routing of static nodes based on XMesh is reliable for the data collection from a mobile node.

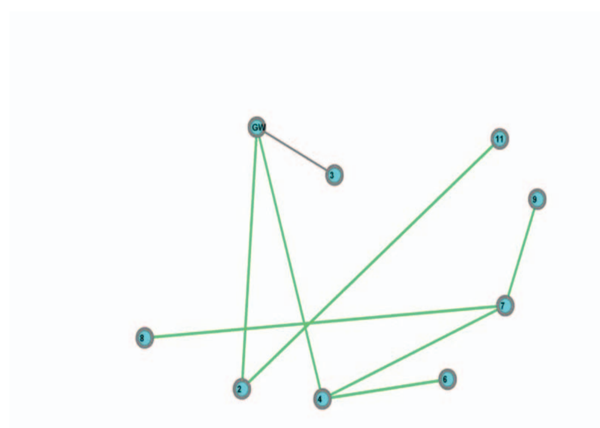

Fig. 7. WSN multi-hop topology with all the nodes working

C. Data fusion in WSN

In WSN based on XMesh protocol, a parent node connecting two or more child nodes has to forward lots of extra packets to the gateway. This leads to energy drain of the parent node. To avoid such extra communication overhead, dynamic data aggregation technique is proposed in the present work.

XMesh protocol is a dynamic routing protocol and hence parent nodes are altered frequently based on the link quality parameter. Based on the present topology, the gateway sends a command to the parent node having more number of child nodes to perform the data aggregation. *XCommand* interface provided by XMesh protocol is used to meet this purpose. After receiving the *XCommand* from the gateway, the parent node intercepts the data it is forwarding from its child nodes. The *intercept()* interface of the XMesh protocol is used instead of the *Receive()* interface.

D. Statistical analysis in Moteview

Various statistical properties of the collected data by a sensor mote can be exploited by plotting a histogram. In Moteview, by selecting the nodes and parameter (light, temperature etc.) histogram can be plotted as shown in Figure 8.

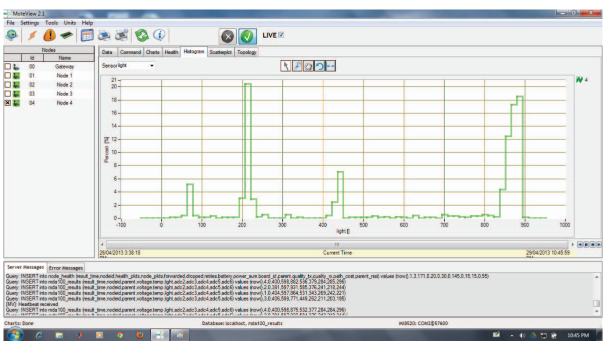

Fig. 8. Histogram of light intensity perceived by node 4

E. Energy consumption analysis

The static nodes deployed in the network make use of XMesh protocol for routing the data towards the BS. XMesh can be operated in high power (hp) or low power(lp) mode as discussed in the section II. The energy consumption analysis for the *HP* and *LP* modes is given in Table III. The sampling rate and hence packet transmission rate is kept same for both the modes of routing $(2seconds)$. In low power routing, the energy is conserved by reducing the channel listening period of the transceiver. The conservation of energy in the case of low power mode takes the cost of latency in topology formation and its update.

F. The mobile node data

The movement of the mobile node is tracked by using its accelerometer readings. The raw accelerometer readings received by the gateway are shown in Figure 9. When the mobile node is not in the range of the gateway, it passes the data by selecting the parent node from the nearby static nodes present in its communication range. Sometimes, the delay caused by the parent selection process causes certain amount of data loss, and the same is depicted in Figure 9.

Routing Mode	Depletion in voltage level of the node per hour
High power	42.9 mV
Low power	1.33 mV

TABLE III
ENERGY CONSUMPTION ANALYSIS

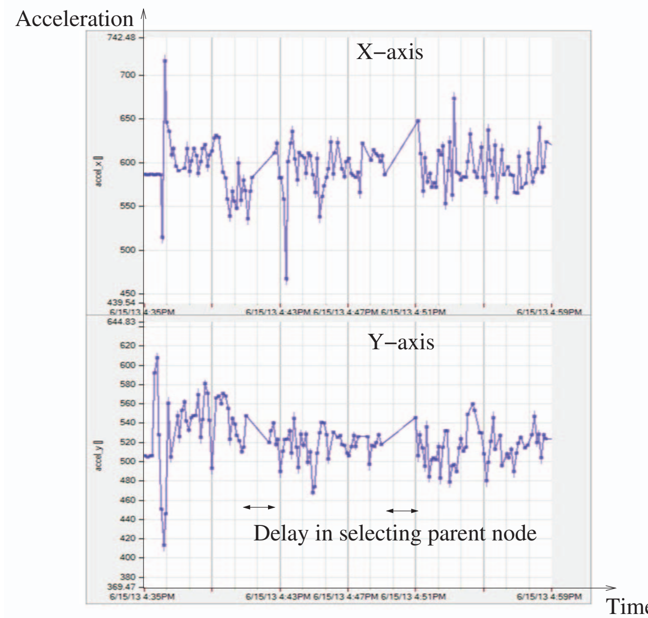

Fig. 9. Raw data collected by two-axis accelerometer

The calibrated accelerometer data, in terms of acceleration due to gravity is shown in Figure 10. The magnitude calculation assists in detecting sudden movements of the mobile node.

VI. CONCLUSION

In this paper, the problem of tracking a mobile object is considered. The WSN setup, with the multi-hop routing protocol, as a solution to the problem is discussed. An experimental analysis is carried out in this work, proves the feasibility of the proposed WSN architecture in real world applications. The depletion in the battery voltage level of a node corresponding to the routing mode of the Xmesh is given. The low power mode can be chosen when the object is moving slowly. When a node is made to operate its Xmesh in high power mode, the node depletes its energy sooner. In order to compensate partially, data dynamic aggregation is used. The data from the mobile node is made available to the gateway, even when the mobile node is not in the radio range of the gateway. However,

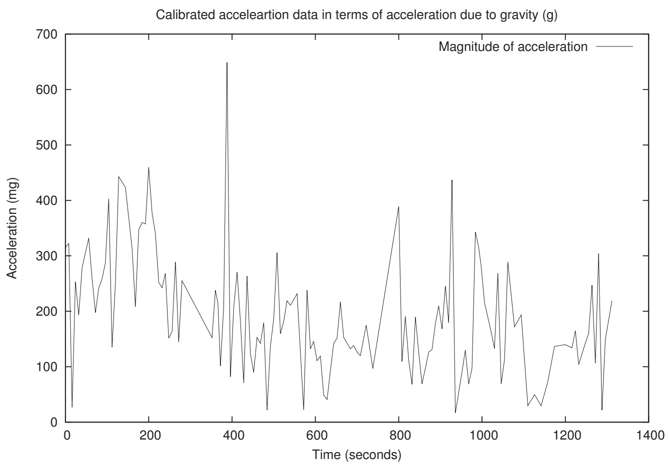

Fig. 10. Acceleration of the mobile sensor node in terms of acceleration due to gravity ($g = 9.82m/s^2$)

it has been observed that the process selecting the parent by the mobile node causes some time delay and consequently resulting certain data loss. A novel approach based on the portable computer, *Raspberry Pi*, is proposed and verified for providing the WSN data to the authenticated remote user. The paper provides the generalised framework for the WSN applications with Internet connectivity.

REFERENCES

[1] K. Sohraby, D. Minoli, and T. Znati, *Wireless sensor networks: technology, protocols, and applications*. Wiley-interscience, 2007.

[2] A. Swami, Q. Zhao, Y.-W. Hong, and L. Tong, *Wireless Sensor Networks: Signal Processing and Communications*. Wiley, 2007.

[3] D. Angela, M. Ghenghea, and I. Bogdan, "Supporting environmental surveillance by using wireless sensor networks," in *Electrical and Electronics Engineering (ISEEE), 2010 3rd International Symposium on*, 2010, pp. 216–219.

[4] A. Harun, M. F. Ramli, L. Kamarudin, D. Ndzi, A. Y. M. Shakaff, A. Zakaria, and M. Jaafar, "Comparative performance analysis of wireless rssi in wireless sensor networks motes in tropical mixed-crop precision farm," in *Intelligent Systems, Modelling and Simulation (ISMS), 2012 Third International Conference on*, 2012, pp. 606–610.

[5] R. Yerra, A. Bharathi, P. Rajalakshmi, and U. Desai, "Wsn based power monitoring in smart grids," in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2011 Seventh International Conference on*, 2011, pp. 401–406.

[6] T. Kuroiwa, M. Suzuki, Y. Yamashita, S. Saruwatari, T. Nagayama, and H. Morikawa, "A multi-channel bulk data collection for structural health monitoring using wireless sensor networks," in *Communications (APCC), 2012 18th Asia-Pacific Conference on*, 2012, pp. 295–299.

[7] V. Kumar and S. Tiwari, "Performance of routing protocols for beacon-enabled ieee 802.15.4 wsns with different duty cycle," in *Devices and Communications (ICDeCom), 2011 International Conference on*, 2011, pp. 1–5.

[8] O. Adewumi, K. Djouani, and A. Kurien, "Rssi based indoor and outdoor distance estimation for localization in wsn," in *Industrial Technology (ICIT), 2013 IEEE International Conference on*, 2013, pp. 1534–1539.

[9] E. Upton and G. Halfacree, *Meet the Raspberry Pi*. Wiley, 2012.

[10] X. U. Manual and D. Revision, "Crossbow technologies," *Milpitas, CA, Apr*, 2007.

[11] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[12] T. W. Davis, X. Liang, M. Navarro, D. Bhatnagar, and Y. Liang, "An experimental study of wsn power efficiency: Micaz networks with xmesh," *International Journal of Distributed Sensor Networks*, vol. 2012, 2012.

[13] A. Brokalakis, G.-G. Mplemenos, K. Papadopoulos, and I. Papaefstathiou, "Resense: An innovative, reconfigurable, powerful and energy efficient wsn node," in *Communications (ICC), 2011 IEEE International Conference on*, 2011, pp. 1–5.

[14] A. Teo, G. Singh, and J. McEachen, "Evaluation of the xmesh routing protocol in wireless sensor networks," in *Circuits and Systems, 2006. MWSCAS'06. 49th IEEE International Midwest Symposium on*, vol. 2. IEEE, 2006, pp. 113–117.

[15] S. Waharte, R. Boutaba, Y. Iraqi, and B. Ishibashi, "Routing protocols in wireless mesh networks: challenges and design considerations," *Multimedia Tools and Applications*, vol. 29, no. 3, pp. 285–303, 2006.

[16] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.

[17] M. Conti and S. Giordano, "Multihop ad hoc networking: The reality," *Communications Magazine, IEEE*, vol. 45, no. 4, pp. 88–95, 2007.

[18] C. Mica, "Micaz, and iris motes."

[19] Z. Alliance, "Zigbee specification," *ZigBee Document 053474r13*, pp. 344–346, 2006.

[20] N. Bin Shafi, K. Ali, and H. Hassanein, "No-reboot and zero-flash over-the-air programming for wireless sensor networks," in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012 9th Annual IEEE Communications Society Conference on*, 2012, pp. 371–379.