International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.1 (2014), pp.133-142
http://dx.doi.org/10.14257/ijmue.2014.9.1.13

A Framework for Wireless Sensor Network Fault Rectification

Ahmad Haboush[1], Mihir Narayan Mohanty[2], Binod Kumar Pattanayak[3*]
and Motassem Al-Tarazi[4]

[1]Department of Computer Science, Jerash University, Jerash, Jordan
[2]Department of Electronics & Communication Engineering, ITER, Siksha 'O' Anusandhan University, Bhubaneswar, Odisha, India
[3]Department of Computer Science and Engineering, ITER, Siksha 'O' Anusandhan University, Bhubaneswar, Odisha, India
[4]Department of Computer Science, Iowa State University, USA

Ahmad_ram2001@yahoo.com, mihirmohanty@soauniversity.ac.in, binodpattanayak@soauniversity.ac.in, altarazi@iastate.edu
**Corresponding author*

Abstract

Wireless sensor networks (WSNs) have gained worldwide attention in recent years. These are of great significance in resolving many real-world problems, and have attracted increasing research interests in recent years. In this paper, we consider a hybrid sensor network with fifty sensor nodes, where a subset of the nodes has movement capability, possibly at high energy expense. A static sensor node can check its energy status and seek for replacement when its current energy falls below a certain threshold. If a redundant static sensor is located close to a dying sensor and can fulfill the coverage requirement, it can be used for substitution. One way to repair the fault is to find redundant nodes to replace faulty nodes. Redundant nodes are placed in the position where the average distance to each sensor node is the shortest. Simulation results show that our algorithm can find the proper redundant node in the minimum time and reduce the relocation time with low message complexity. We investigate the dynamics on random geometric networks. Fixed threshold policy is considered that enables the sensor node to vary and continue its service so that data loss can be minimized. This policy is also considered for energy measure of the node. Work has been carried out for faulty node among intermediate nodes. The rectification is done by replacing the faulty node and to continue the network data transfer.

Keywords: *Sensor Network; NS-2; Energy of Node; Threshold policy; Packet.*

1. Introduction

Advance technologies in wireless communication have insisted the development of wireless sensor networks. A wireless sensor network consists of a large number of mobile sensor nodes; those are capable of observing the environment and communicate the information about the environment. Also the sensors are spatially distributed to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, humidity, and to cooperatively pass their data through the network to a main location. The sensor network includes both wired sensor network and wireless sensor network. These are of very small, low-cost, low-power, and capable of processing data and communicating each other. Thus the networks have been utilized in a wide range of potential applications like,

ISSN: 1975-0080 IJMUE

field of medicine, environment exploration, military surveillance, *etc.* A basic sensor network is shown in Figure 1.

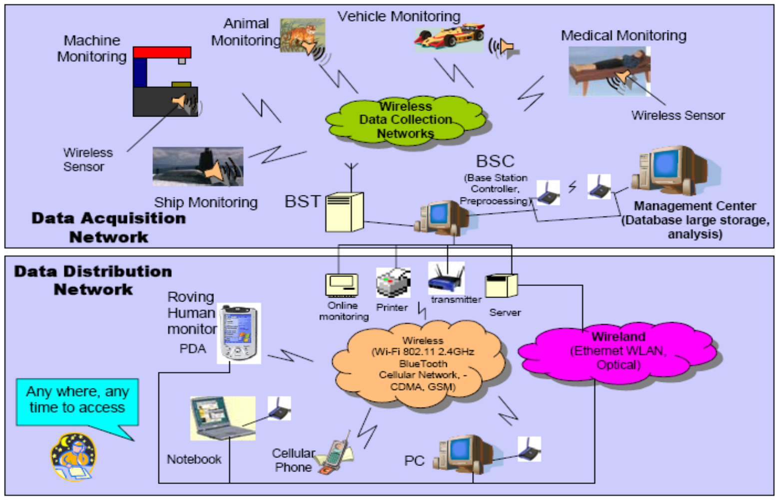

Figure 1. Wireless sensor network

A typical sensor node consists of the building blocks like, power unit, communication unit, processing unit and sensing unit as shown in Figure 2. Due to the low-cost and the mobile computational capability of sensor nodes, they are usually deployed in the harsh or the human-unreachable environment to perform the sensing task. However, there is much accidental damage in the harsh environment. Because of the environmental interference and the low-power essence of sensor nodes, sensor nodes are prone to failure unexpectedly. The faulty node introduces many errors into the network and corrupts the network. It may be in presence of *faulty* sensor measurements. These events usually span some geographic region and in many application scenarios, the detection of the event boundary may become more important than the detection of the entire event region [1-4].

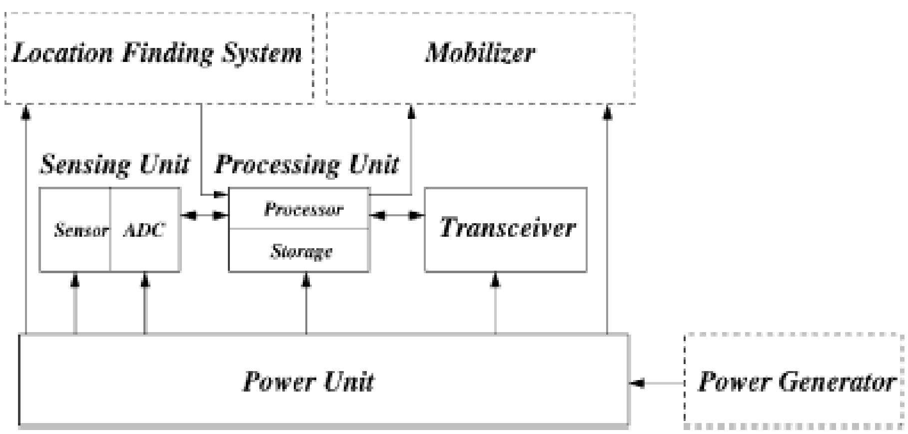

Figure 2. The components of a sensor node

The primary function of wireless sensor networks is to gather sensor data from the monitored area. Due to faults, the sensor data may not be collected or collected data might be wrong. Hence it is important to detect events in the presence of wrong sensor readings and misleading reports. Each sensor node makes a decision on the fault status of itself and its neighboring nodes. Sensor nodes have limited battery power without recharging capabilities. Nodes running out of power may cause topology changes in sensor networks even without mobility. New sensors with fresh batteries may be injected to a sensor network, already in use, to enhance and ensure its correct operation. They are battery operated and thus their performance tends to deteriorate as power is exhausted.

Sensor failure may cause network topology changes and in extreme cases, network partitioning and the network may fail to deliver its functionalities required by applications. For accurate sensing, this faulty sensor node should be replaced immediately. One of the simple solutions for this problem is to have redundant deployment and to maintain a set of backup nodes. When a sensor node fails, backup node is used to replace it. However, if the redundant sensor is a mobile node and is located far away from the dying sensor, a protocol should be in place to locate redundant mobile sensor and to schedule the movement of mobile sensor for replacement purposes.

Section-2 describes the related literature. Section-3 gives the experimental setup. Section-4 explains the methodology. Section-5 shows the results obtained. Section-6 concludes the work.

2. Related literature

It has been reported the related work in the area of fault detection and recovery in wireless sensor networks (WSNs). Authors in [4-6] have proposed fault detection schemes for wireless sensor networks. They use centralized, distributed, or hierarchical models. As the fault models, for detection of faults, such as transient and permanent faults, are used. In [7], an overview of existing outlier detection schemes has been proposed for wireless sensor networks. Sensor readings that appear to be inconsistent with the remainder of the data set are the main target of the detection. The idea here is to depend on neighbourhood monitoring of the nodes. Every sensor node monitors its surrounding and whenever a transmission signal is detected by a sensor node, it would check if the signal strength of the transmitting node is compatible with the originator node's geographical position. Even though this approach is applicable, it is not efficient in many ways. The large overhead needed for transmitting data is a problem both for sending and processing. Also it is not energy efficient since all nodes are monitoring and processing data all the time [8].

Atakli *et al.*, presented a malicious node detection scheme using weighted trust evaluation for three-layer hierarchical network architecture [9]. They are updated depending on the distribution of neighboring nodes. An improved intrusion detection scheme based on weighted trust evaluation was proposed in [10-11]. The mistaken ratio of each individual sensor node is used in updating the trust values. The procedure to detect malicious node by comparing its output with an aggregation value is reported in [8-9].

Some techniques have been proposed by authors for fault detection, fault tolerance and repair in sensor networks. A survey on fault detection through fault tree analysis and conversion of fault tree to Bayesian network is included in [4-7] that also suggests that fault detection can be also found by detecting the faulty node by cut-set generation method. Fault detection has been done by End-to-End measurements and sequential Testing in WSN. To improve network connectivity, additional relay nodes have been placed in wireless sensor networks [12]. Implementation of routing protocol and simulation in WSN by NS-2 software has been also analyzed. Energy efficiency and maximum lifetime routing for extending the life time is described in [13]. Here, Fault tolerance in Internet such as network availability and performance has been discussed by authors. Also, Hierarchical and cluster-based approaches for fault detection and repair have been proposed by researchers in [14]. In our work, simulation is performed to detect and replace the faulty nodes. It is approached by determining the nodes with lower energies than the threshold energy. This data is then provided to the simulation for a fault free network.

3. Experimental Setup

A wireless sensor network is established within a boundary of 1000 X 1000 m^2. It has Two-Ray-Ground propagation, drop-tail queue, omni-directional antenna. Two routing protocols such as AODV and DSDV have been tested and the simulation runs for 105s. The initial energy is provided as 100 Joules. The nodes have been assigned transmitting power of 0.250W and receiving power of 0.250W by using the energy model. All the mobile nodes have their initial node position assigned to them. A total of 10 redundant nodes have been placed. All the nodes are attached to the agents determining the source and destination nodes. The source and sink nodes are then attached for the data flow. Thereafter, the traffic is attached to the agent and is provided with start and stop time. Then, finally the simulation is run in NS-2.

4. Methodology

The sensor network fails to achieve its objectives when it cannot provide the desired coverage. In order to avoid the network partition and the coverage hole, it is necessary to find the redundant node to replace the faulty node as soon as possible. This process is called the *sensor relocation*. The sensor relocation consists of two stages. The first stage is to find the nearby redundant node in the sensor network. The second stage is to relocate the redundant node to replace the faulty node.

Every node gets the localization information from the neighboring nodes and also computes the localization information itself and compares these two values. If the difference is small enough, that node decides that there is no adversary around causing the localization problem in its location.

4.1. Redundant nodes arrangement algorithm

In order to find the redundant node as soon as possible, the redundant nodes arrangement algorithm arranges the deployed redundant nodes to the specific position to form redundant walls. In general, if the distance between redundant walls and the faulty node is shorter, the time to find the redundant node will be less. In addition, since each sensor node in the sensor network may fail, we desire to arrange redundant nodes to a proper position where the average distance from the redundant node to each sensor node is the shortest.

There are n sensor nodes in a row and the position of each sensor node is from 1 to n. The distance between each pair of sensor nodes is R, and we deploy the redundant node in the position x. Therefore, if the redundant nodes arrangement algorithm can arrange redundant nodes to the center of the sensing field, the time to find the redundant node will be the shortest. After redundant nodes arrive at the center of the sensing field, redundant nodes will send messages to notify neighboring sensor nodes regarding the existence of redundant nodes. Specifically, redundant nodes which move to the center will send messages to notify neighboring sensor nodes.

4.2. Faulty nodes replacement algorithm

The procedure of the faulty nodes replacement algorithm is described as follows. All sensor nodes periodically send hello messages to neighboring sensor nodes to verify whether they are alive. If one of the neighboring sensor nodes does not reply, other neighboring sensor nodes conceive that the sensor node that did not reply is failed. To replace the faulty node, the sensor node which found the faulty node performs two steps and as follows.

(1) Finding the redundant node and

(2) Replacing the faulty node.

Threshold-based model considers the dynamics on a network of interacting agents, each of which must decide between two alternative actions and whose decisions depend explicitly on the actions of their neighbors according to a simple threshold rule [15-16].

We define the energy consumption during a successful global cascade. The total energy consumption in a wireless sensor network is, in most part, due to communication between nodes, computing, and storage (neglecting some smaller miscellaneous costs). The communication is the main part of energy consumption, and is directly related to the network structure and dynamics, while energy used for computing can be regarded as a constant; so it can be easily evaluated. Consequently, we only consider the energy consumption in communications between sensor nodes.

According to the nature of sensor nodes, the wireless broadcast is the most commonly used method for local communication. Thus, the energy cost for local broadcast is proportional to the square of radio range R, *i.e.*, El = cR2, where c is a coefficient that we scaled to 1. Here we propose algorithm with threshold techniques:

(i) Assign 50 units to each sensor node with some additional redundant nodes.

(ii) Assign no. of packets to transfer and the corresponding power is Pmax

(iii) Enter the threshold value (Thvalue).

(iv) Node senses the packet.

(v) If power<= Thvalue then transfer the packet to next most probabilistic node by searching another nearest redundant node.

(vi) If search==true then transfer the packet to that replaced node.

(vii) Else transfer the packet to any node available in neighbouring list.

The trace file contains all the information of the events happening within the simulation. So we extract the energy information from the trace file. If the energy is less than the threshold, then the node is considered as faulty node. The coordinate and time of the faulty nodes is obtained using the following algorithm:

Simulation process consists of three steps.

Step 1: Simulation design:

Figure 3 shows the configuration of a network under consideration. The network consists of five nodes n0 to n4. In this scenario, node n0 sends constant bit rate (CBR) traffic to node n3, and node n1 transfers data to node n4 using a file transfer protocol (FTP). These two traffic sources are handled by transport layer protocols: User Datagram Protocol (UDP) and Transmission Control Protocol (TCP), respectively. In NS2, the transmitting object of these two protocols is a UDP agent and a TCP agent, while the receivers are a Null agent and a TCP sink agent, respectively.

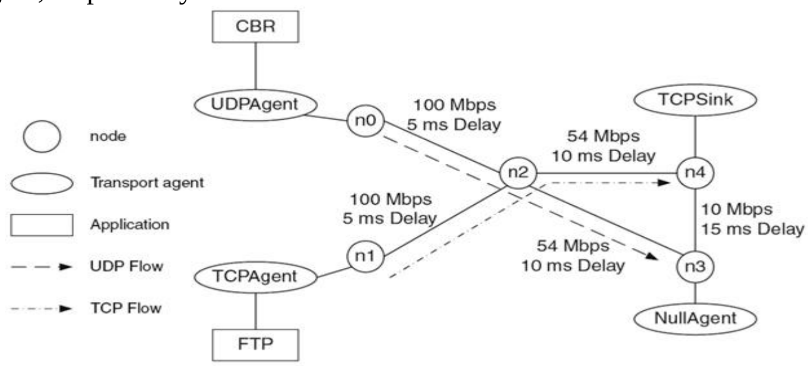

Figure 3. A sample network topology

Step 2: Configuring and Running Simulation

Initially a program creates a simulator instance. It creates a trace file and a NAM trace file, respectively. It defines procedure finish*{}* . Finally, it creates nodes and links them together. The Simulator is created in the program by executing "new Simulator". The returned Simulator handle is stored in variable ns. The variables myTrace and myNAM are the file handles for these two files, respectively. One informs NS2 to collect all trace information for a regular trace and other for NAM trace, respectively. The procedure finish*{}* is invoked immediately before the simulation terminates. The keyword *global* informs the Tcl interpreter that the variables ns, myTrace, myNAM are those defined in the global scope (*i.e.*, defined outside the procedure). It must flush the buffer of the packet tracing variables. Next, it has to close the file associated with handles myTrace and myNAM. Then it executes the statement "nam out.nam &" from the shell environment. Finally, NS2 has to exit with code 0. Each pair of nodes should be connected with a bi-directional link using an instproc duplex-link *{*src dst bw delay qtype*}* of class simulator, where src is a beginning node, dst is an terminating node, bw is the link bandwidth, delay is the link propagation delay, and qtype is the type of the queues between the node src and the node dst. It is similar to the instproc duplex-link*{...}*, or create a uni-directional link using an instproc simplex-link*{...}* of class simulator. Finally, it sets the size of the queue between node n2 and node n3 to be 40 packets.

Step 3: Post Simulation Processing–Packet Tracing:

Packet tracing records the detail of packet flow during a simulation. It can be classified into a text-based packet tracing and a NAM packet tracing.

a)Text-Based Packet Tracing

Text-based packet tracing records the details of packets passing through network checkpoints (*e.g.*, nodes and queues). A part of the text-based trace obtained by running the above simulation (myfirst_ns.tcl) is shown below.

5. Discussion of Results

With various time instant, the results have been noted. Initial stage represents the WSN communication in Figure 4. After some time, Figure 5 and Figure 6 represent the gradual reduction of energy that will be the faulty node and is marked as YELLOW color node. It seems to be less than the threshold value. In Figure 3, the same node has no energy for data transfer and is represented by RED color. Next to it in Figure 7, the replacement has been occurred. Finally in Figure 8, the graph is shown for energy reduction with respect to time. The work is simulated using NS-2 environment and the figures have been represented as the network animator output.

5.1. Replacement of a dying node with a mobile node

Here we are having 4 sensors. Out of them, 3 are static sensors & 1 is dynamic sensor. In this program, 3 sensors are transmitting data among them & the energy of sensor is fixed to 30 joules & simulation time is 120 sec. The program is written for both AODV and DSDV protocol. In this program, one of the static node dies, *i.e.*, its energy becomes zero, and on that occasion, a new sensor which is a dynamic one replaces this node and maintains the communication link. NAM editor o/p at different instants is given below [Figure 4, Figure 5 and Figure 6].

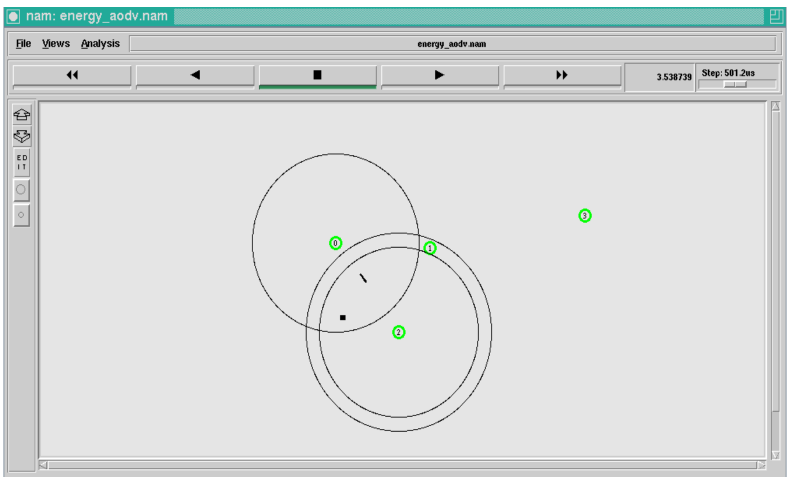

Figure 4. Simulation output at t=3.5 sec

At this stage the node 0 is communicating with the node 1 and node 2. Node 3 is at ideal state but still its energy is decaying.

Figure 5. Simulation output at t=83 sec

At this stage the energy of node 0 falls below the threshold and node 3 will be replacing the dying node.

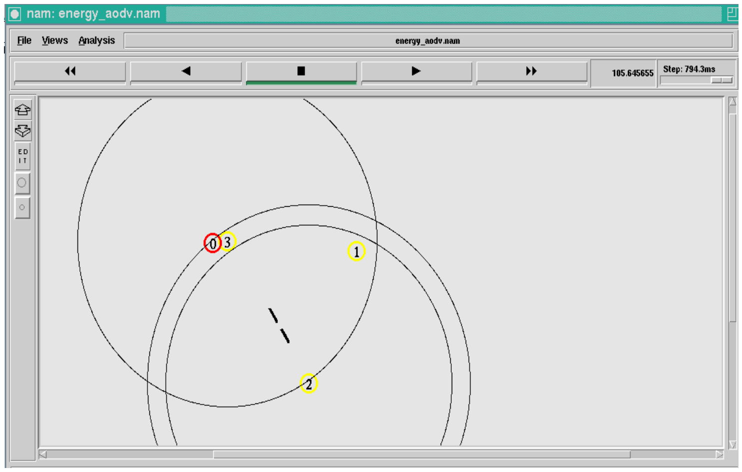

Figure 6. Simulation output at t= 105 sec

At this stage, finally node 3 replaces the node and starts communicating with node 1 and node 2.

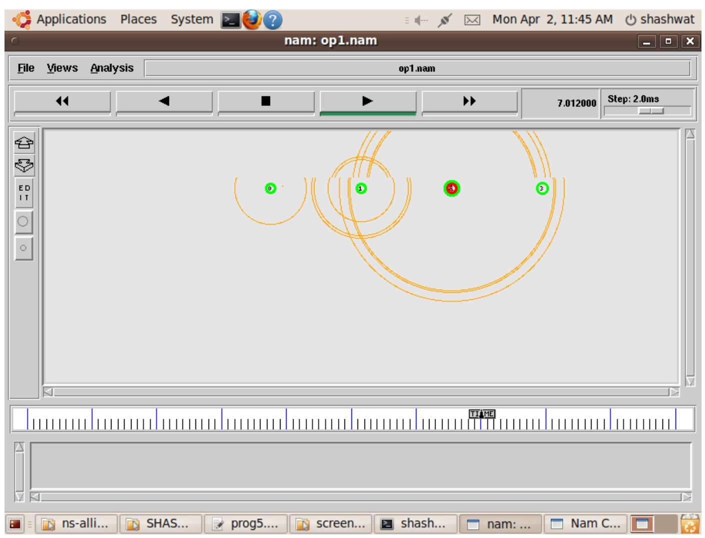

Figure 7. Replacement of the faulty node at t=7.0 sec

X-graph is an open file format used to describe a graph object. X-graph is a general purpose x-y data plotter that can plot multiple data. The final result is plotted in the x-graph.

Here we have found a relation between various parameters (like delay, throughput and energy) and number of nodes. The idea behind this is to relate, when some nodes fail or die out, how the performance of the sensor network gets affected. For delay and throughput, we have written ns2 tcl codes for various nodes and the trace file is extracted. From the trace file, again the delay and throughput parameters are extracted for each number of nodes and this was plotted with respect to the respective number of nodes. Here a single tcl code is used for 12 nodes.

5.2. Energy performance in replacement of nodes

In AODV, when a static node dies out at that time, the mobile node starts functioning, applies its algorithm and estimates to reach to the dying node and hence replaces that. Therefore the energy of mobile node comes into picture only when it starts to replace a dying out static node [Figure 8].

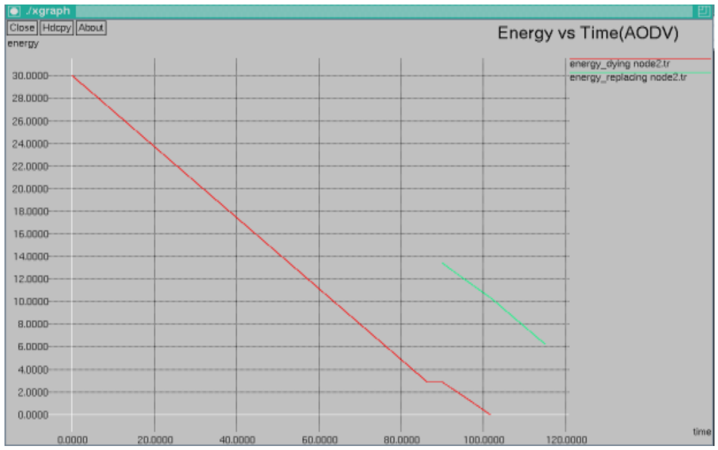

Figure 8. Energy Level in Time for AODV

In case of DSDV protocol, the mobile node is functional always and when a static node dies out, it applies the algorithm and estimates to replace that node. Hence the mobile node is consuming energy from the beginning of the simulation. That means the mobile node also needs a continuous energy supply like the static node for its functioning for the whole period, and from the energy utilization point of view it is similar to static node [Figure 9].

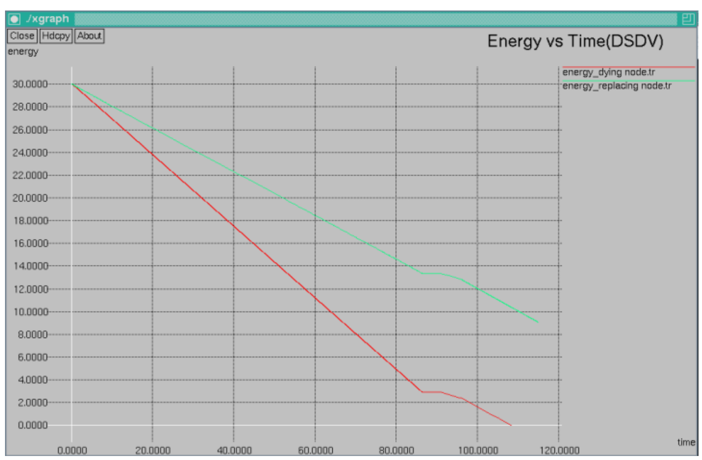

Figure 9. Energy Level in Time for DSDV

6. Conclusion

Efficient power consumption is a challenging problem in battery-powered wireless sensor networks. The fact is that each node has a limited battery power and it is impossible or infeasible to recharge the batteries. Therefore, it further reduces power consumption and increases the network lifetime. The network lifetime is directly proportional to the efficient power consumption, thus dysfunctional of any node causes serious damage to the network service considering nodes dual role of data originator and data router. Simulation results show that the lifetime of a sensor node capable of working continuously by using threshold policy. It also reduces idle time period and power loss during idle period. If number of service rates offered by a node are more than two then instead of using single adaptive threshold policy, multiple fixed thresholds are needed to select required service rate. A sensor node with multiple service rates is much complex than that with two service rates and also switching power and switching time overheads increases. So it is a better option to achieve the long lived wireless sensor networks having sensor nodes offering service.

References

[1] M. N. Mohanty and B. Mohanty, “An Approach for Faulty Node Replacement in Wireless Sensor Network”, IJCA Special Issue on Int. Conf. on Computing, Communication and Sensor Network CCSN 2012, no. 2, **(2013)** March, pp. 32-35.
[2] H. S. Hu and G. H. Qin, “Fault Management Frameworks in Wireless Sensor Networks”, 4th International Conference Intelligent Computation Technology and Automation, Shenzhen, **(2011)** March 28-29, pp. 1093-1096.
[3] M. H. Lee and Y. -H. Choi, “Fault Detection of Wireless Sensor Networks”, Computer Communications, vol. 31, no. 14, **(2008)**, pp. 3469-3475.
[4] A. Akbari, N. Beikmahdavi, A. khosrozadeh, O. Panah, M. Yadollahi and S. V. Jalali, “A Survey Cluster-Based and Cellular Approach to Fault Detection and Recovery in Wireless Sensor Networks”, World Applied Sciences Journal, vol. 8, no. 1, **(2010)**, pp. 76-85.
[5] T. Le, N. Ahmed and S. Jha, “Location-free fault repair in hybrid sensor networks”, First International Conference on Integrated Internet Ad hoc and Sensor Networks, **(2006)** May 30-May 31.
[6] R. Duan, J. Tu and D. Dong, “A Novel Hybrid Approach of Fault Tree and Bayesian Networks for Fault Diagnosis”, Journal of Computational Information Systems, vol. 6, no. 11, **(2010)**, pp. 3605-3612.
[7] B. Wang, W. Wei, H. Dinh, W. Zeng and K. R. Pattipati, “Fault Localization Using Passive End-to-End Measurements and Sequential Testing for Wireless Sensor Networks”, IEEE Transactions on Mobile Computing, vol. 11, no. 3, **(2012)** March.
[8] W. Junior, T. Figueriredo, H. -C. Wong and A. Loureiro, “Malicious Node Detection in Wireless Sensor Networks”, the 18th International Parallel and Distributed Processing Symposium (IPDPS’04), **(2004)** April 26 – 30, Santa Fe, Nex Mexico, USA.
[9] I. M. Atakli, H. Hu, Y. Chen, W. -S. Ku and Z. Su, “Mali-cious Node Detection in Wireless Sensor Networks Using Weighted Trust Evaluation”, Proceedings of Spring Simulation Multi-Conference, Ottawa, **(2008)** April 14-17, pp. 836-843.
[10] L. Ju, H. Li, Y. Liu, W. Xue, K. Li and Z. Chi, “An Im-proved Detection Scheme Based on Weighted Trust Evaluation for Wireless Sensor Networks”, Proceedings of the 5th International Conference on Ubiquitous Information Technology and Applications, Sanya, **(2010)** December 16-18, pp. 1-6.
[11] M. Momani and S. Challa, “Survey of Trust Models in Different Network Domain”, International Journal Ad Hoc, Sensor & Ubiquitous Computing, vol. 1, no. 3, **(2010)**, pp. 1-19.
[12] X. Han, X. Cao, E. L. Lloyd and C. -C. Shen, “Fault-Tolerant Relay Node Placement in Heterogeneous Wireless Sensor Networks”, IEEE Transactions on Mobile Computing, vol. 9, no. 5, **(2010)** May.
[13] G. Anastasi, M. Conti and M. Di Francesco, “Extending the Lifetime of Wireless Sensor Networks Through Adaptive Sleep”, IEEE Transactions on Industrial Informatics, vol. 5, no. 3, **(2009)** August.
[14] A. Akbari, N. Beikmahdavi and M. Mohammadi, “A New Algorithm Fault Management by Clustered in Wireless Sensor Network”, World Applied Sciences Journal, vol. 12, no. 10, **(2011)**, pp. 1784-1790.
[15] C. Callegari, R. G. Garroppo, S. Giordano and G. Nencioni, “A New Markov Model for Evaluating the ALR Dual-Threshold Policy”, 2nd International Workshop on Green Communications (GreenComm2), **(2009)**, Honolulu, Hawaii, USA; pp. 1-5.
[16] C. Gunaratne, K. Christensen and S. W. Suen, “Ethernet Adaptive Link Rate (ALR): Analysis of a Buffer Threshold Policy”, IEEE GLOBECOM 2006, San Francisco, CA, **(2006)** November.