Dynamic Load Balancing Protocol (DLBP) for Wireless Sensor Networks

Hamzeh Aljawawdeh, Iman Almomani, Member IEEE
Computer Science Department
King Abdullah II School for Information Technology (KASIT)
The University of Jordan, Amman 11942, Jordan
hamzeh@jawawdeh.com, i.momani@ju.edu.jo

***Abstract*—Wireless Sensor Networks (WSNs) are resource-constrained networks that have been applied in many applications. Many researchers have presented routing protocols to prolong the lifetime of WSNs. In this paper we present a Dynamic Load Balancing Protocol (DLBP) that exploits all network nodes to achieve load balancing and prolong the lifetime of WSN. DLBP has succeeded to build a load balanced tree, eliminate the need for control messages during data routing, keep the load of the WSN balanced during data routing, send messages to next hops without route-discovery delay, quickly maintain and fix network errors and failures. Simulation results show that the network success ratio has reached 97%. Routing overhead has decreased by 72% and network lifetime has increased by 20% comparing to other tested algorithms.**

Keywords—Wireless Sensor Networks; WSN; Load Balancing; Routing; Energy Aware;Network lifetime.

I. INTRODUCTION

A WSN is a network of hundreds or thousands of wireless sensor nodes. Each node is a small input device that gathers data by sensing the desired environmental parameters such as heat, humidity and movement. Sensors usually send the collected data to a more powerful node called the sink. The sensor node usually measures a physical quantity and converts it into a signal which can be read by an observer or by an instrument [1].

Sensors could be found anywhere as their size could be small. This feature increased the number of applications that depend on the sensor networks. Even though the sensor signal can reach the sink but most of the designers of WSNs prefer to make the communication between sink and sensors among other sensors such that less energy would be consumed in the communication process [2]. Increasing the number of sensors in WSN allow it to cover wide areas and send more messages to the sink and that could extend the WSN lifetime [3].

Sensor nodes actually have limited resources such as energy, computational power and available storage [4]. The absence of wired power supply and the difficulties in replacing or recharging batteries make the wireless node life short. The communication process consumes more energy, while most of data and information processing take place at the sink most of the time [5]. The main goal of the nodes is to sense and then deliver the gathered information to the sink to do the complex processing. The Random distribution of the sensor nodes in addition to their location that could be in a hostile environment, make such network easy to be reached. Anyone could physically damage the sensors or even listen to the collected data illegally [6].

The rest of this paper is structured as follows. Section II reviews some of the approaches in the literature that handle load balancing in WSN. Section III present Dynamic Load Balancing Protocol (DLBP) that improves and overcomes the exiting protocols in terms of network success ratio, overhead and lifetime. Section IV presents the results and evaluation of the proposed protocol. Section V draws conclusions and presents avenues for future work.

II. LITERATURE REVIEW

Message transmission consumes most of the sensor's energy; hence there exist many studies that focus on finding new techniques to route data with the minimum possible effort to save sensors' energy. According to the survey in [7] the way to prolong the lifetime of the WSN is by balancing the load on all nodes at each level, especially those nodes which are very close to the sink. There exist many techniques to achieve load balancing in WSNs. In this section we categorize and discuss some of them.

A. Achieve Load Balancing by Constructing a Balanced Tree

According to [8] the load balancing in WSN is achieved by constructing a load balanced network topology. Balanced Low-Latency Converge-Cast Tree (BLLCT) achieves bottom-to-top load balancing. Every node has many parents. The child chooses one of the candidate parents according to its energy and number of children. This algorithm needs extra processing and more computing energy but still not too complex. The presented idea in [8] is based on AODV-Shortest path Algorithm [9]. In this algorithm the node that has less candidate parents should choose the path to route data before other nodes that have more choices. Also, the heavy

978-1-4799-2303-8/13/$31.00 ©2013 IEEE

loaded nodes that have many children should have more flexibility in choosing the candidate parent. BLLCT depends on constructing a load balanced tree from the beginning.

B. Achieve Load Balancing by Calculating Next Hop Weight

In [10] it is assumed that the lifetime of the network ends when any of its node's energy is completely consumed. Level of node is called node-grade. That means the sink is in the highest grade while its children are in the first grade. If a node wants to send a message to the sink it has to send it to the higher grade node until it reaches the sink. Choosing the next hop depends on the weight of that link, the weight is calculated using the distance (grade) of the node, and the remaining energy. A loop on all candidate nodes takes place to calculate the best next hop. The weight of each node from the neighbors is calculated depending on the distance between the two nodes and energy of the neighbor node. Then the possibility of nodes to be chosen as next hop is getting calculated and the node with the highest possibility will be selected as the next hop. Calculating the weight of the next hop just before sending a message is the way used in many protocols such as [11] and [12]. The problem is the generated delay before sending every new message.

C. Load Balancing Techniques Inspired from Animals Behavior

Studying the behavior of animals encourage researchers to develop creative ideas that are inspired from these animals. [13] Presents a routing protocol that was inspired from ant's behavior. Ants usually attempt to find best path to their home. New paths to the sink are generated on demand, and data about routes are stored in sensor nodes.

D. Load Balancing Techniques based on Clustering the Network Nodes

Clustering means to create small groups of neighbor nodes and call it cluster. Each cluster is connected to the network through one of its nodes that is called cluster head. In [14] a clustering technique is used with sleeping periods for cluster heads to make the network balanced and prolong its lifetime.

It can be concluded from the above approaches that we need to introduce an integrated protocol that benefits from the strength points of the existing protocols such as building a balanced tree in the first phase and keep updating data about neighbors. On the other hand, this protocol should overcome the weaknesses by removing the delay before sending messages, eliminating control packets, and keep updating the load on all network nodes until reaching stability stages. These are the main goals of presenting a Dynamic Load Balancing Protocol (DLBP) for WSN as will be explained in the following section.

III. DYNAMIC LOAD BALANCING PROTOCOL

A. Overview

Considering the challenges in WSN, mainly the limited resources of sensor nodes in addition to the distribution nature of the sensor nodes, this paper presents a Dynamic Load Balancing Protocol (DLBP) for WSN that aims to prolong the WSN lifetime. DLBP takes into considerations the network topology construction, and data routing. In addition to the load balancing technique used in this protocol, data filtration metrics were exploited to reduce the redundant data and improve the performance of WSN. The way used by DLBP is to adjust the used paths dynamically while routing data. DLBP consists of three main phases; tree construction phase, data filtration phase and dynamic load balancing phase.

B. Tree Construction Phase

Finding at least one path from every node to the sink is the main goal when constructing the tree topology. Some enhancements on [1] were achieved to give better performance and remove the unneeded messages while constructing the tree. The number of children for any parent is limited to some number to give other nodes at the same level better chance to get children. The enhancements include removing several control messages such as *Unready*, *New Node, Request Parent, Change ID* and *Inform* messages. Also the cost of link between nodes was added to the header of control messages.

C. Data Filtration Phase

During this process the neighbor nodes which almost reside in near area could sense the same event. This could cause redundant data to be sent to the sink. Reducing the sent data could effectively prolong the WSN lifetime, as transferring data consumes most of the node's energy. DLBP filters data that come from different children or close neighbors and drop redundant messages. This process should take place when the parent node makes sure that the data is redundant and the location of data is the same. DLBP uses limited historical data records in the parent node to compare data comes from children. Data records that are compared together are only the records come from the same area in a limited period of time and then redundant messages should be removed. The data filtering takes place at the application layer. If two neighbor nodes from the same level generate two packets at the same time, then DLBP sends these two data packets to the application layer. The application reads and compares the piece of information that is saved in each data packet and drops the redundant one. This phase is not a standalone phase; it is actually part of the data routing phase as will be explained in the next section.

D. Dynamic Load Balancing Phase

This part discusses the presented idea for a routing technique that should consider load balancing while transferring data from source to destination. DLBP provides a technique that keeps the load balancing of the WSN even after sensors start collecting data from the environment and sending them to the sink. Most of the existed techniques that apply load balancing in WSNs, consider load balancing at the beginning or at the topology construction phase assuming that the sensing rate is equal for all nodes. DLBP attempts to remove complex computations and prevent delaying messages before being sent.

Nodes in the DLBP could play selfishly by refusing to participate in routing packets to save their energy. If node does that it would conserve its energy, while nodes that involve in transmission the packets will reduce their lifetime. DLBP attempts to find the best utilization such that most nodes should participate in data transmission process. Larger number of nodes that participate in data transmission process could be better to find more alternative paths to the sink and help in achieving load balancing in the network.

DLBP in the first phase of tree construction attempts to find at least one main path from any node to the sink. On the other hand, the neighbors table is implemented to find alternative paths to the sink. Each path to the sink should have a cost. Lower cost path is better to be used by the node. The link cost between any two nodes has an initial value which could be increased once it is needed. The cost of routing data from the parent node to the sink is accumulated to the cost between the child node and its parent. If the node has many options it will take the best option and use it unless a change on its cost has taken place. Fig. 1 shows the cost calculation in DLBP in a simple way. Initially, link cost is set to one and total cost is set to the number of hops toward the sink (logical distance). The cost will be updated later according to the loads.

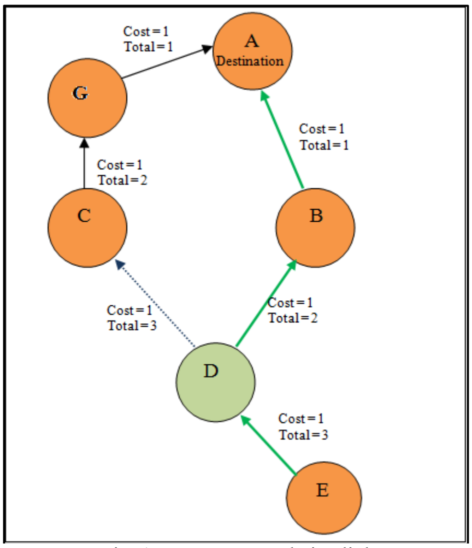

Fig. 1: DLBP accumulative link cost

When a child is engaged to a parent it will accumulate the initial cost to the parent cost such that the total cost is the actual cost that is sent to the neighbors and children. If the node has more than one path with different costs it will choose the path with the minimum cost to be the default path. In Fig. 1 note that node D has two paths to the sink; one through node B with cost 2, and another one through node C with cost 3. Node D chooses node B to be the default next hop until a new change takes place on the whole path which forces node D to make a new check and find the best path again.

The node energy and traffic of messages play a significant role in changing the cost of any path. Every node has at least one basic path to the sink that passes through its parent. If the load increases on some path while other paths at the same level have fewer loads then DLBP increases the cost of the overloaded path. Hence, nodes attempt to find a list of alternative paths with less cost and use them. Fig. 2 shows an example of using alternative paths when cost is being changed. Node D is the source node of the sent data packet and node A represents the destination node. Node A is the parent for C and B. Node B is the parent for nodes E and D. The solid lines represent the basic paths between children and their parents while dashed lines represent alternative paths between the node and its neighbors. The cost represents the path weight and less cost means better path to be used. In Fig. 2 (A), source node sends its packets through its parent B because it has the lowest cost, so the path will be D-B-A.

In Fig. 2 (B) the cost of link B-A has been increased; therefore, the links D-B and E-B have increased their costs as well. Changing the cost gives node D more options to route its data. Now it can use the path through node B or through node E alternatively. In the neighbors table of node D, the nodes B and E are flagged to be the best next hop, and all packets from node D should be routed through these two options. In Figures 2 (C) and (D) other changes take place on the cost, therefore, routing path alternatives keep changing dynamically.

Load balancing techniques usually work from top to bottom; because nodes in top levels have wider view to the network, more information about the tree and data movements than lower levels or leaf nodes. Therefore taking any decision related to load balancing, that may affect any level of nodes, should be taken from nodes in higher level. DLBP applies this higher level monitoring technique, which gives it more strength comparing to other techniques. In DLBP the root node monitors the whole tree while every sub tree is monitored by its head, so changing the link cost is not centralized in the root of the network.

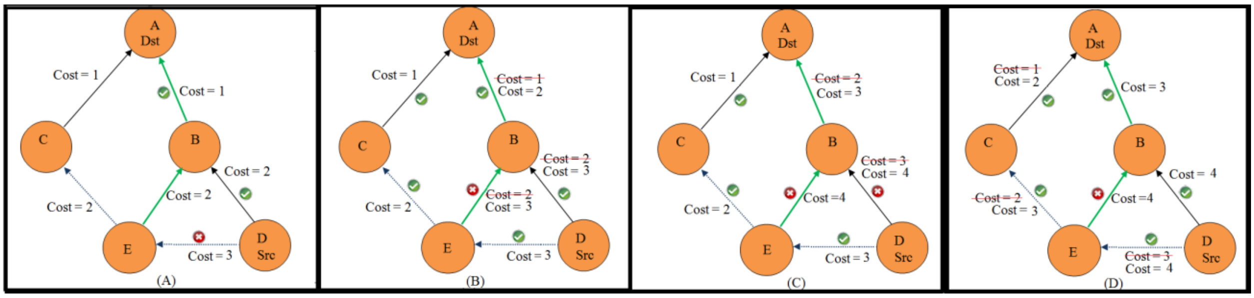

Fig. 2: Example of path choosing based on links cost. The next hop is the link with minimum cost. Note that ☑ represents the paths with lowest cost, and ☒ represents the paths with higher costs that will not be selected unless there is no other option. Note the ☑ and ☒ after each cost change.

In DLBP each head node of any sub tree has to monitor its children and directly connected neighbors. Monitoring the children and neighbor nodes is achieved by accumulating the number of sent messages. When a branch of the tree makes a high load on its parent, the parent node increases the cost to enforce that branch to reduce its messages to become in the normal range. Controlling sub branches is done by controlling the head of that sub branch. To control the amount of data that could be received from a node, the following points should be considered:

a. Cost of links between nodes is the value that forces the node to find the path with the lower cost. DLBP directs the load from the overloaded paths by increasing their costs and consequently direct data packets to find lower cost paths.

b. DLBP replaced control packets by adding small values to the data message header. If the cost of some link is changed, then neighbor nodes should be informed about that change. DLBP does not send special control packet to inform neighbor nodes about the change in the link cost. Instead of that, the new cost is added to the header of data packets, and nodes that receive that data packet will check this header and update the cost if needed.

c. After many updates on the cost and changes in the paths the network will reach a stability situation. That means a slow adjustment process is taking place on the network to make it well-balanced while sending data packets.

In DLBP, every node has at least one path to the sink, and it may have some alternative paths through its neighbors. The data messages are usually sent through the path with the lower cost. If there are many paths that have the same cost then many options would be available and this would distribute the load on multiple paths. DLBP keeps watching the paths and then increase the cost of any high loaded path such that nodes which use this path would start seeking for alternative paths. By doing this, the load will be distributed on all available paths and will not be focused on the basic one. On the other hand, to reduce the routing overhead, DLBP extends the header of data messages to be used instead of sending control messages. DLBP adds the current energy level to the header of data messages. Nodes that receive this data message can update the energy of that node using the piece of information in the header. The extra fields that are added to data packet and used by DLBP are the *Cost* and *ToBeAlerted* fields such that each field of them reserves 4 bytes only.

E. Link Maintenance and Termination

Every routing protocol should always have alternative plans to fix errors or link failures in the network. DLBP provides the nodes with all alternative paths in case there is a link failure or a dead node. DLBP protocol handles dead node failure by controlling the links that lead to that node. If the node's energy reaches a low value, it should inform neighbor nodes to use other alternative paths, this is called link termination, whereas, the terminated link will be canceled and nodes will not use it anymore.

Link termination could be implemented by changing the cost to a negative value instead of the current cost. When neighbor nodes find a negative value in the cost they will immediately stop routing packets through the failed path. Again, there is no need to use special control packets to inform neighbors with the link failure, and no need to search for new paths because all the alternatives paths should be available in neighbors table. This will make the network maintenance very fast, without any additional control packets or loads.

IV. RESULTS AND EVALUATION

In this section the simulation results of DLBP are discussed and evaluated. The Network simulator Omnet++ version 4.2.2 was used to conduct the simulation experiments with Castalia framework version 3.2. Number of nodes was between 50-300 nodes with transmission range of 50m. Radio TX Level was -5 dBm and terrain size was 1000 m^2. DLBP is compared to the protocol presented in [10] which is called An Energy-Balance Routing Algorithm Based on Node Classification for WSNs. It will be called Xue algorithm for the rest of the paper. There are some reasons behind choosing Xue algorithm to compare with DLBP: Xue algorithm is a dynamic load balancing technique, which is a common point between Xue algorithm and DLBP and it is one of the recent researches in this field.

Also the authors have simulated their algorithm and the simulation shows good results comparing to some of widely used algorithms which are Short-Path routing algorithm [11], STRP and the power aware version of STRP which is called STRP-PA [12]. Xue algorithm is also easy to be implemented and the similarities between DLBP and Xue make them easy to be compared with each other.

The comparison metrics used here are the Network success ratio, routing overhead and network lifetime.

A. Network Success Ratio

The network success ratio is evaluated by calculating the number of sent and received packets from each node. Received packets ratio gives an important indicator about the ability of DLBP to deliver the sent messages. Fig. 3 shows the received packets ratio, which is the percentage between the received packets to the total sent packets.

$$\text{Success Ratio} = \frac{\text{Received Packets}}{\text{Total Sent Packets}} \qquad (1)$$

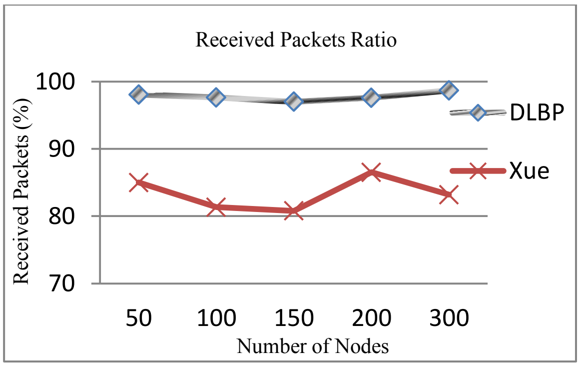

Fig. 3: Received packets ratio vs. Number of nodes

The reason behind the high percentage of received packets ratio and the low percentage of failed packets ratio comparing to other protocols is the lack of interference and overlap between nodes. The data filtration plays a significant role in reducing the interference. If a node in the neighborhood receives a packet that belongs to another node then it should drop it. Choosing the best next hop and use less loaded paths reduces the interference and increases the packet reception rate of the network as well.

B. Routing Overhead

Control packets are used to construct the network, inform neighbor nodes that a node is ready to have children, make sure that sent data packets were successfully delivered to the next hop, and many other possible uses. Routing overhead can be defined as the ratio of total number of sent control packets to the total number of data and control packets sent from the node i.e.

$$\text{Routing overhead} = \frac{\text{Total Sent Control Packet}}{\text{Total Sent Data \& Control Packets}} \qquad (2)$$

DLBP reduces the using of control packets; they are used only in the first phase which is the tree construction. Instead of sending a special control packet to inform neighbor nodes to take some action or update information about neighbors, just few flags were added to the header of data packets to do the same task.

Xue's algorithm still has more control packets to update the neighbors table, such as updating the energy level of node. The elimination of control packets while routing data in DLBP removes the overhead of control packets which is called the routing overhead. Note that the distribution of nodes on the field area could play an important role of minimizing the number of the sent control packets. If there are too much nodes in a small area, the number of sent control packets could be increased, that because node receives more ready messages, and it could fail many times at the beginning to be engaged to some parent. On the other hand, parent nodes could reach the full number of children which means reject all next engagement requests.

In Fig. 4 the routing overhead is shown by calculating the ratio of control packets to the total number of sent packets. Note the very small percentage of control packets in DLBP. The control packets that were found in DLBP are used only when the tree is getting constructed for the first time. Number and distribution of nodes plays a very important role in increasing or decreasing sent control packets. If there are many nodes in the same area then one or two attempts to send engagement message could be enough to have a parent. Thus, no need to send much more packets as there are many options in the neighborhood.

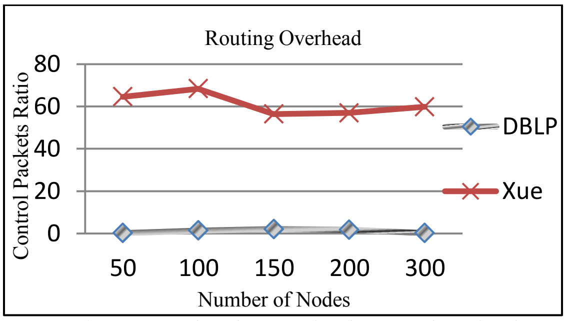

Fig. 4: Control packets overhead vs. Number of nodes

C. Network Lifetime

The goal of presenting DLBP is to increase the network lifetime. To evaluate network lifetime a simulation scenario was tested for twenty hours on a network of 100 nodes. Fig. 5 shows the network lifetime. The x-axis is the simulation hours and y-axis is the remaining energy in Joule (in thousands). It can be noticed from Fig. 5 that nodes in DLBP remain stable all the time, and consume energy monotonically. While in Xue protocol nodes consume energy faster after running the simulation for longer time. The reasons behind this enhancement are the following:

a. Filtering redundant data reduces sent and received data packets, which saves more energy and results in prolonging network lifetime.

b. Eliminating control packets while routing data reduces energy consumption and prolongs network lifetime.

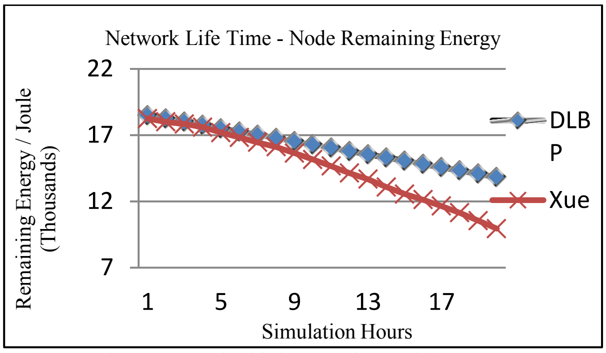

Fig. 5: Network Lifetime. Node remaining energy vs. Simulation hours

V. COLCLUSIONS AND FUTURE DIRECTIONS

This paper has presented a Dynamic Load Balancing Protocol (DLBP) which is an energy aware routing protocol for WSNs that uses load balancing techniques to prolong the network's lifetime.

DLBP has three phases. In the first phase DLBP constructs a tree with at least one path to the sink. Then in the second phase DLBP applies filtering technique to reduce the number of messages and to remove redundant data packets. In the data routing phase DLBP eliminates the control packets and balances the load on the network. As a result for these three phases, DLBP has prolonged the WSN lifetime by 20% in comparison to Xue's algorithm. The routing overhead has also decreased by 72%, and the network success ratio has increased by 16%.

Three related directions should receive attention in the future. Firstly, building a mathematical model for DLBP. Secondly, simulating network total throughput and thirdly, considering security issues for DLBP.

REFERENCES

[1] Iman Almomani, Maha Saadeh, Hamzeh Al Jawawdeh, Mousa Al-Akhras, "Energy Awareness Tree-Based Routing Protocol for Wireless Sensor Networks", The 10th WSEAS International Conference on Applied Computer and Applied Computational Science (ACACOS '11), 8-10 March, 2011, Venice, Italy, pp. 26-30.

[2] Jamal N. Al-Karaki and Ahmad E. Kamal, "Routing techniques in wireless sensor networks: a survey", Wireless Communications, IEEE, vol. 11, issue 6, Dec. 2004, pp. 6–28.

[3] Ning Xu, "A Survey of Sensor Network Applications", IEEE Communications Magazine, vol. 40, University of South Carolina, 2002.

[4] Imad Jawhar, Nader Mohamed, Dharma P. Agrawal, "Linear wireless sensor networks: Classification and applications", Journal of Network and Computer Applications, Volume 34 Issue 5, September, 2011, pp. 1671-1682.

[5] Amulya Ratna Swain, R. C. Hansdah and Vinod Kumar Chouhan, "An Energy Aware Routing Protocol with Sleep Scheduling for Wireless Sensor Networks", Advanced Information Networking and Applications (AINA), 24th IEEE International Conference, 20-23 Apr 2010, Perth, WA, pp. 933-940.

[6] Renita Machado, Sirin Tekinay, "A Survey of Game Theoretic Approaches in Wireless Sensor Networks", Computer Networks vol. 52, issue 16, 13th Nov 2008, pp.3047-3061.

[7] Dipak Wajgi, Nileshsingh V. Thakur, "Load Balancing Algorithms in Wireless Sensor Network : A Survey", IRACST – International Journal of Computer Networks and Wireless Communication, 2012.

[8] Tzu-Ping Chung, Joe-Jiang, "A Load Balancing Algorithm Based on Probabilistic Multi-Tree for Wireless Sensor Networks", IEEE 5th Conference on Sensing Technology, 28th Nov – 1st Dec 2011, Palmerston North, pp. 527-532.

[9] F. Tong, W. Tang, L.-M. Peng, R. Xie, W.-H. Yang, and Y.-C. Kim, "A node-grade based AODV routing protocol for wireless sensor network,"Proc. Second International Conference on Networks Security, Wireless Communications and Trusted Computing, 2010, vol. 2, pp. 180-183.

[10] Jilong XUE, Xiaogang QI, Chenyu WANG, "An Energy-Balance Routing Algorithm Based on Node Classification for Wireless Sensor Networks", Journal of Computational Information Systems, vol. 7, Jul 2011, pp.2277-2284.

[11] Jie Gao and Li Zhang, "Load-balanced short-path routing in wireless networks", Parallel and Distributed Systems, IEEE Transactions, vol. 17, issue 4, April 2006, pp. 377-388.

[12] Gong Ben-Can, Li La-Yuan, Jiank Ting-Yao, Xu Shou-Zhi, "Distributed Spanning Tree-Based Routing Protocol for Wireless Sensor Networks", Microelectronics & Computer, 2008

[13] Ahmed M. Shamsan Almshreqi, Borhanuddin Mohd Ali, Mohd Fadlee A. Rasid, Alyani Ismail and Pooria Varahram, "An Improved Routing mechanism using Bio-Inspired for Energy Balancing in Wireless Sensor Networks", In Proceedings of ICOIN, 2012, pp. 150-153.

[14] Dipak Wajgi and Nileshsingh V. Thakur, "Load Balancing Based Approached to Improve lifetime of Wireless Sensor Netowrk", International Journal of Wireless & Mobile Networks, vol. 4, issue. 4, 2012, pp. 155-167.