نموذج ترخيص

أنا الطالب : حمزة جميل يوسف الجوادوة أمنح الجامعة الأردنية و/ أو من تفوضه ترخيصاً غير حصري دون مقابل بنشر و / أو استعمال و / أو استغلال و / أو ترجمة و / أو تصوير و / أو إعادة إنتاج بأي طريقة كانت سواء ورقية و / أو إلكترونية أو غير ذلك رسالة الماجستير / الدكتوراه المقدمة من قبلي وعنوانها.

Energy Balanced Topology for Wireless Sensor Networks

وذلك لغايات البحث العلمي و / أو التبادل مع المؤسسات التعليمية والجامعات و / أو لأي غاية أخرى تراها الجامعة الأردنية مناسبة، وأمنح الجامعة الحق بالترخيص للغير بجميع أو بعض ما رخصته لها.

اسم الطالب: حمزة جميل يوسف الجوادوة

التوقيع: حمزة جوادوة

التاريخ: ٢١/٥/٢٠١٣ م

ENERGY BALANCED TOPOLOGY FOR WIRELESS SENSOR NETWORKS

By
Hamzeh Jamil Aljawawdeh

Supervisor
Dr. Iman Musa Almomani

This Thesis was Submitted in Partial Fulfillment of the Requirements for the Master's Degree in Computer Science

Faculty of Graduate Studies
University of Jordan

May, 2013

This Thesis (Energy Balanced Topology for Wireless Sensor Networks) was Successfully Defended and Approved on 9th May 2013

Examination Committee	**Signature**
Dr. Iman Musa Almomani (Supervisor) Associate Professor of Wireless Networks and Security	------------------------------
Dr. Wesam Almobaideen Associate Professor of Computer Networks	------------------------------
Dr. Basel Ali Mahafzah Associate Professor of Parallel, Distributed Computing & Interconnection Networks	------------------------------
Dr. Wail Mardini Associate Professor of Computer Science Jordan University of Science and Technology	------------------------------

DEDICATION

I lovingly dedicate this thesis to my family and my friends who offered me unconditional love and support each step of the way. In particular, the patience and understanding shown by my parents and my wife is greatly appreciated. Thank you all, without you this degree would remain only a dream.

ACKNOLEDGMENT

First of all thank ALLAH for the wisdom and perseverance that he has been bestowed upon me during this research project, and indeed, throughout my life

I would like to express my sincere gratitude to my advisor Dr. Iman Almomani for the continuous support of my master's study and research, for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor for my study.

Besides my advisor, I would like to thank Dr. Wesam Al Mobaideen, for his guidance and with whom I have first started learning the computer networks.

I would like also to thank all the professors at the University of Jordan for their support and for everything I have learned from them. I would like to thank my colleagues for encouraging and supporting me all the time.

Last, but not least, I would like to thank my wife for her understanding and love during the past few years. Her support and encouragement was in the end what made this dissertation possible. My parents receive my deepest gratitude and love for their dedication and the many years of support during my undergraduate studies that provided the foundation for this work.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACK	Acknowledge Message
ADC	Analog to Digital Converter
ADV	Advertise Message
ANT-B	Ant Backward
ANT-F	Ant Forward
BAN	Body Area Networks
BLLCT	Balanced Low-Latency Converge-Cast Tree
BS	Base Station
CH	Cluster Head
DLBP	Dynamic Load Balancing Protocol
EBRA	Energy Balanced Routing Algorithm
EOFS	Environment and Observation Forecasting System
FEAR	Fuzzy-Based Energy Aware Routing Protocol
GEAR	Geographic and Energy Aware Routing
GPS	Geographic Positioning System
ID	Identifier
LEACH	Low Energy Adaptive Clustering Hierarchy
MAC	Medium Access Control
REQ	Request Message
SPIN	Sensor Protocols for Information via Negotiation
TDMA	Time Division Multiple Access
TEEN	Threshold sensitive Energy Efficient sensor Network protocol
TX	Transmit
WSN	Wireless Sensor Network

ENERGY BALANCED TOPOLOGY FOR WIRELESS SENSOR NETWORKS

By
Hamzeh Jamil Aljawawdeh

Supervisor
Dr. Iman Musa Almomani

ABSTRACT

Wireless Sensor Networks (WSN) is currently one of the hot topics for researchers. The limited resources of sensor nodes, such as battery, memory and processor, create a big challenge to researchers. One of the challenges when working on sensor nodes is to overcome the limited resources to create a routing protocol that saves energy and prolong the WSN lifetime.

This research presents a Dynamic Load Balancing Protocol (DLBP) for WSN. The presented technique was inspired from Game Theory. It works dynamically to balance the load on all WSN nodes and exploits the network nodes to distribute the load fairly on every available sensor node.

Xue algorithm is a dynamic load balancing technique (Xue, et al., 2011). Xue algorithm main idea is to calculate the weights of all neighbor nodes then find the possibility for each link. The data packets will be sent through the link with the highest possibility.

A set of simulation experiments were conducted to evaluate the presented protocol through different metrics. Network scalability was studied, and it was found that DLBP saved energy with a ratio reached 20% comparing to Xue algorithm. The success ratio reached 97%, which is 16% better than Xue algorithm. Moreover, the routing overhead decreased by 72%, and the complexity of calculations decreased by 99.9993%. The network lifetime also increased by 20%.

CHAPTER 1

Introduction

1.1 Overview

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 then send the collected data to a more powerful machine 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. Sensors are hardware devices that produce measurable response to a change in a physical condition like temperature and pressure. Sensors are used to sense or measure physical data of the area to be monitored (Almomani, et al., 2011, a).

Figure 1 shows a typical sensor node structure; it shows the sensor main parts which are the transceiver, microcontroller, analog to digital converter (ADC), power unit, and external memory. The transceiver is the part that handles sending and receiving data, it consists of a transmitter and a receiver. The second part is the microcontroller which processes data, performs tasks and controls other parts of the sensor node. Analog to Digital Converter (ADC) is that part which converts continuous analog signals to digital signals or numbers (Wikipedia, 2013).

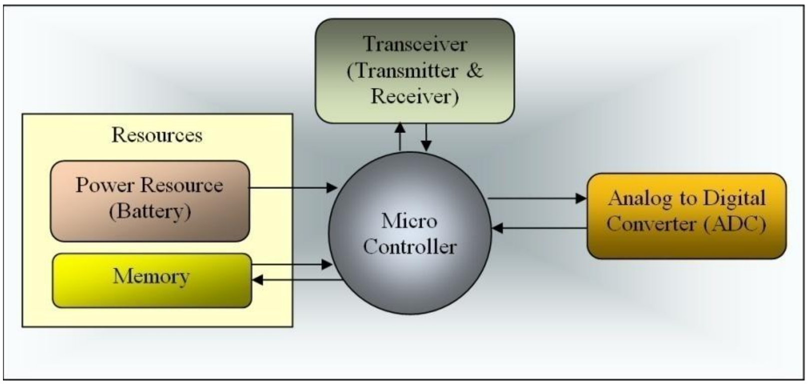

Figure 1: A typical sensor node structure (Wikipedia, 2013)

Sensors could be found anywhere as their size could be small and tinny, and this feature increased the number of applications that depend on the sensor networks. Figure 2 shows some sensor types that we may see every day in our lives. Figure 2 shows some small and tinny sensor devices that could be used in some applications (Harvard, 2013).

Figure 2: Small Sensor Nodes (Harvard, 2013) (Singularityhub, 2013)

1.1.1 Wireless Sensor Network

A WSN almost consists of two main things: the first one is the base station (BS) or sometimes it is called the Sink; which is a powerful device that controls and receives signals from wireless sensors in the network (Almomani, et al., 2011, a). The sink also can make some computations and evaluation on the accumulated data (AlKaraki and Kamal, 2004). The second thing in the WSN is the sensors; the WSN consists of hundreds or thousands of sensor nodes that are used to gather data and communicate directly with the sink or among other sensors to deliver the needed data (Ning, 2003).

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 a less energy would be consumed in the communication process (AlKaraki and Kamal, 2004). When the WSN contains more sensors it could be extended to cover a wide area, and the large number of sensors in the WSN should extend the network life such that it could live more and send much more messages to the sink (Ning, 2003).

1.1.2 Applications of WSNs

WSNs are widely used in many applications such as monitoring applications, military applications, medical care, Environment Observation and Forecasting Systems (EOFS) (AlKaraki and Kamal, 2004). Some of these applications could be very critical or sensitive such that any fault in the WSN may endanger the lives of some people or animals or put them under risk. Therefore any mistakes or weaknesses in the WSN topology or routing algorithm could be very expensive (Ning, 2003).

WSNs are commonly used in civil or military applications, and these applications could be classified into data collection and surveillance that includes object tracking as a special case. Another classification is to classify applications into event driven and periodically data collecting applications (Singh, et al., 2010).

For data collection hundreds or thousands of sensors may be spread in some field or area to gather some information about the environmental changes. These sensors collect data periodically. In some systems the collected data is aggregated and analyzed to help in keeping the condition under control. As an example of using WSNs in data collecting is spreading sensors in a field to watch humidity, heat, pollution and many other environmental parameters (Jawhar, et al. 2011) (Swain, et al., 2010).

Volcano alert system is another example on using WSN on the volcano. WSN could be used to give an early alert if there is an active volcano by sensing the earthquakes and temperature in the volcano location (Jawhar, et al. 2011).

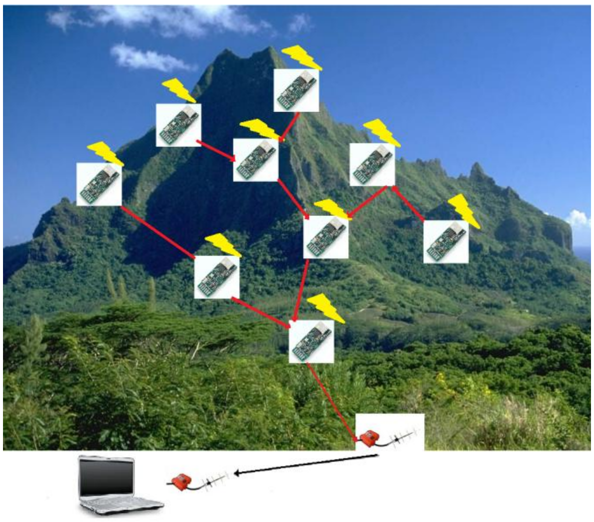

Figure 3: Volcano alert system (MCSL, 2013)

Monitoring and surveillance applications require spreading or planting sensors in fixed locations such that we can know exactly the place of the event. Data sensing in these application depends on some event so we can call these systems event driven systems. Applications of this type could be used indoor or outdoor. For example they could be used inside a building to monitor the movements or track a person who is walking through the building rooms (Swain, et al., 2010). Forest fires alert system is another example of using WSN. It was found that early alerts could help to stop fires before they grow up. Sensor networks were one of the best alerting systems for such applications. The idea is to plant hundreds of sensor nodes in each forest with some sink that are connected to remote monitoring stations through satellite connection points (wireless access to work stations).

If a fire started in some place in the forest the close sensor immediately sends a signal that shows the exact place of fire such that users of the system can take a fast action to stop fires. See figures below.

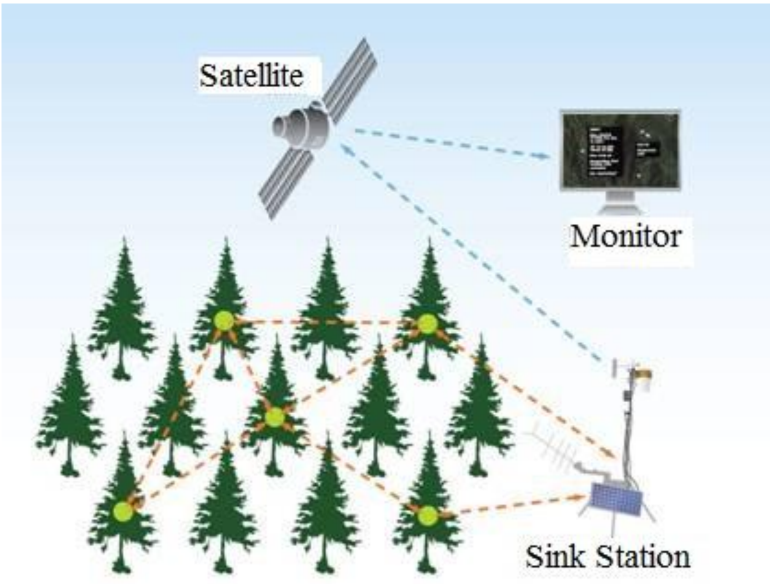

Figure 4: Fire alert system in forest using WSN (Altenergymap, 2013)

1.1.3 Current Challenges of WSNs

Researchers who are working with WSNs usually face many challenges. The small size of sensor nodes causes some restrictions in the energy and storage. The large number of sensor nodes in the network is another challenge. This section explains some of these challenges in more details.

Some challenges that face the researches are caused by the small size and limited resources of sensor nodes. Sensor nodes actually have limited resources because of their small size, and also the available resources are affected by the small size such as the energy, computational power and available storage as shown shortly (Jawhar, et al. 2011).

Limited energy of sensor nodes appears because of the small size of the node, plus the absence of wired power supply, so it was found that sensor nodes are battery-driven most of the time. It is difficult to replace or recharge node's batteries, because there are hundreds or thousands of sensor nodes most of the time, and the sensor network environment could be hostile or remote area. The energy in the sensor node is used to make computations or data communication; but actually the communication process consumes more energy, while most of data and information processing take place in the sink most of the time (Swain, et al., 2010). The main goal of the nodes is to sense and then deliver the gathered information to the main work station and let it do the complex processing.

Storage space at the sensor node is used to save gathered data or received data from other sensors until sending them. The problem appears when there are too much nodes in the network, and the sensor node is asked to store the keys of every single node in that network, which is impossible. So the challenge here for the researchers is to provide techniques that reduce the usage of sensor memory resources (Singh, et al., 2010).

The network should be scalable and flexible to the enlargement of the network's size. The design of routing protocols should take in account that the number of nodes in the network could be extended without affection the performance of the communication over the network, in other words the network should be kept stable even with the increasing number of nodes in that network. Another target should be taken in account when developing a new routing protocol for WSN is increasing the tree life with the increased number of nodes in that tree; this may looks easy or something come by the way, but this is not true; because the increasing number of nodes means more communication messages, more fault tolerance work, more energy to construct and fix the tree … etc (Singh, et al., 2010).

As shown before, most of the time the sensors are distributed randomly on the target area without taking care of the place of each sensor. The challenge face the WSN designer is to make it possible to construct the topology.

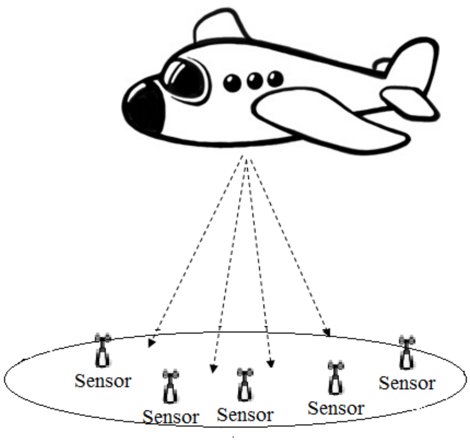

Figure 5: Ranodm nodes distribution. Sensor nodes could be distributed randomly and thrown from a plane (BWN, 2013)

Hostile environment is another challenge to overcome. Sensor networks can be deployed in remote or hostile environments such as battlefields. This means that these sensors could be reached and accessed physically by anyone who could damage the network or even listen to the collected data illegally (Machado and Tekinay, 2008).

Sensors usually are connected through wireless channels. Due to the limited bandwidth of the sensor some problems may arise such as interference between nodes and some messages could be lost before being received.

Sensors in the same area most of the time sense the same data hence we get many redundant messages. These redundant messages increase the load on the network when trying to send to the sink, and consume computational energy when trying to analyze them (Machado and Tekinay, 2008).

1.2 Problem Statement

Energy efficient routing protocol is a major concern in the WSNs. Therefore, the network lifetime and energy consumption has been considered in the presented work. The energy consumption should be reduced to prolong the WSN lifetime. These objectives can be achieved when routing data within WSN is handled properly along with the battery life of the sensor nodes.

More precisely, a sensor node consists of small units such as microcontroller, memory, radio unit, sensing unit and power supply unit consisting of non-rechargeable, non-changeable battery. The small size of typical sensor nodes causes some limitations and restrictions and shortens the WSN lifetime. The issue of this research is to overcome the WSN limitations and prolong the network lifetime.

1.3 Research Contribution

The primary focus of this thesis is development and evaluation of a presented protocol to prolong the lifetime of WSN by applying load balancing on the network nodes. By considering the sensors limitations and studying the literature of other techniques that take care of load balancing in WSN we are looking for a solution to overcome the weaknesses in other techniques.

It is very important to take the challenges of WSNs into account to reduce energy consumption and to prolong the network lifetime. Many studies try to prolong the network lifetime by minimizing the energy consumption while doing several tasks in WSN. This thesis research aims to present a better way to reduce the energy consumption in sensor nodes while constructing the network topology or routing data during the network execution.

Building the network topology with the minimum possible control messages is the first step to reduce energy consumption, therefore extending the network lifetime (Almomani, et al., 2011, a). Transferring data through the WSN consumes most of nodes energy; so that building an energy awareness routing protocol would certainly help to prolong the WSN lifetime (Almomani, et al., 2011, b & c). Sending some nodes in sleeping periods while there is no data to be sent is another additional technique that was used to save nodes energy. However, there are many other techniques that prolong the WSN lifetime by saving energy; such as minimizing the size of sent data by using compression methods to compress data, using data cleaning, and to make sure that destination nodes do not have the same data before the sending process.

In this research, a Dynamic Load Balancing Protocol (DLBP) is presented. Our technique has been successfully prolonged the network lifetime comparing to other related work. DLBP reduces the consumed energy and successfully makes the network more scalable. DLBP has three phases; the first one is the tree construction phase, then the second phase is the data filtering phase, next is the third phase which is the data routing phase. After completing the tree construction phase, data filtering and then data routing phases are getting started. In DLBP the control packets were eliminated completely after the first phase, which reduces the routing overhead on the network. Moreover, a filtering process is achieved on the data packets to drop redundant packets. Dropping redundant data packets reduces the interference between packets and reduces node consumed energy as well. On the other hand, the calculations to find best routes to the sink were reduced too. Finding next hop became easier and faster and with no delays. DLBP is evaluated using Omnet++ simulator with Castalia framework. Our simulation results show that DLBP is better than other related algorithms in terms of the evaluation metrics.

1.4 Thesis Organization

This thesis is organized as follows. First, a brief introduction is given to clarify the subject of this thesis and to discuss the problem formulation, goal of this thesis and the thesis contribution.

Next, Chapter 2 provides a summary of related work and studies that compliment the study in this thesis. First it discusses the different categories of WSNs and how they were classified. Then it digs more deeply into more related work by discussing the use of load balancing in other researches to prolong the WSN lifetime.

Chapter 3 introduces the presented approach of this thesis. It explains the network construction phase. Then it discusses the data filtration method in the presented protocol. Then the presented protocol for dynamic load balancing is discussed in more details.

Chapter 4 presents the simulation results. Also these results are evaluated and discussed in details. Then chapter 5 recommends some future directions.

CHAPTER 2

Literature Review

WSNs are usually classified into three main categories regarding the network topology; flat-based structure, hierarchical-based structure and location-based structure. Sensor nodes in each of the mentioned network structures play different roles and WSN use different routing techniques (Swain, et al., 2010). Also the way of dealing with sensor nodes differ from one topology to another as will be shown shortly.

2.1 Flat Based Networks

In flat networks all sensor nodes play the same role. Most of the time the sensor node knows only its neighbors and there is no global identifier for each node because of the large number of nodes in such networks (AlKaraki and Kamal, 2004). The flat topology networks usually use centric routing algorithms; in which the base station broadcasts a request to some region and waits for a response from sensors located in that region. Transferring data through the network take place either in flooding manner or using data-centric routing. Flooding means that all nodes get the request and have data replays to their neighbors, so many redundant data occurs. While in data-centric routing data is identified using attribute based naming because there is no global node identity.

Sensor Protocols for Information through Negotiation (SPIN) (Heinzelman, et al., 1999) is an example of flat-based routing protocols. In SPIN the sensor node that has data broadcasts a small-sized message called advertise message (ADV) to its neighbors (AlKaraki and Kamal, 2004). Neighbor nodes will reply to the sender with request message (REQ) unless they have the same data. If the source node got a REQ reply from some nodes it will send the data to these nodes. This process is called negotiation, and it should reduce the redundant data with the minimum possible number and size of messages. The ADV message is usually very small but contains meta-data about the real message that would be sent (Kulik, et al. 2002). SPIN protocol uses ADV messages in order to make sure in advance that target nodes in the next hop do not have this piece of information from another place.

Scalability and simplicity are the main advantages of the flat-based routing protocols (Singh, et al., 2010). On the other hand the nodes around the sink will lose their energy sooner than other nodes in the network because of the traffic on them. Drawbacks of flat-based routing protocols mainly caused by duplicating sent messages to the same node (AlKaraki and Kamal, 2004). Overlapping is one of the main drawbacks, it happens when two nodes sense the same region send the same data to the same neighbor. Even by gossiping and sending meta-data to overcome these problems there become a delay in propagation of data through nodes.

2.2 Location Based Routing Protocols

Location-based routing protocols are a special type of routing protocols that require the location information of the sensor nodes to route data (Singh, et al., 2010). Location information could be used to find the distance between nodes such that the energy consumption could be estimated and then reduced. The motivations of developing location based routing protocols are two main things: first, many applications need the node location as a reference address to its site, or some closer place. The second thing is that the location of sensor nodes and distance between them could be used to enhance the efficiency of routing protocols. At first the location based protocols was used for mobile ad hoc networks. However they are more suitable for WSNs in which there is no or less node mobility and the nodes' locations are static (Ning, 2003)(Karthickraja and Sumathy, 2010). Geographic and Energy Aware Routing (GEAR) is a good example on this kind of protocols (Yu, el al., 2001).

Location-based routing protocols or geographic routing is based mainly on geographical information. This type of routing protocols has many drawbacks and weaknesses that related to energy issues (Nurhayati, et al., 2011). One of the major drawbacks is the constraint on network topology from being applied to large scale network. Sensors usually depend on their location to send messages through one hop. Nodes do not maintain neighbors' tables so they communicate directly with the sink. This consumes node's energy because the network is not designed for long distance wireless communication (Nurhayati, et al., 2011). Nodes in this topology are required to posses geographic information which is not easy since the devices that operate with Geographic Positioning System (GPS) consumes a large amount of energy. Even other solutions such as planting the sensors in their place could be expensive and make more limitation of the number of nodes in the network. Location-based routing protocols could be much scalable that routing protocols that maintain routing tables because they deal only with neighbor nodes which saves the energy and memory. Also all nodes are assumed to be able to vary their own transmission range, by adjusting transmit power, but all nodes have the same maximum transmission power (Peng and Kemp, 2011).

2.3 Hierarchical Based Routing Protocols

In Hierarchical-based topology some nodes play different roles to achieve some special tasks. Cluster Head (CH) is an example of node that have special task which is connecting some cluster to the network. Also if we have a tree-based network the head of each sub tree plays the same special role (AlKaraki and Kamal, 2004)(Karthickraja and Sumathy, 2010). Building a hierarchical-based topology even as a tree or clusters provides a good technique to route data through the network (Jinghua, et al., 2009). Moreover constructing a hierarchical-based topology may save the energy of sensor nodes while routing data because the largest effort is done when constructing the network topology (Al Herbawi, et al., 2009). Using unicast and multicast methods instead of broadcasting messages should reduce the energy consumption while transferring data through the network.

One of the most famous hierarchical based protocols is the Low Energy Adaptive Clustering Hierarchy (LEACH) (Heinzelman, el al. 2000). Leach is a cluster-based protocol that divides the network into clusters. Each cluster is a collection of neighbor node with on coordinator node that has extra privilege and called cluster head (CH). The cluster head has the responsibility of creating and manipulating a Time Division Multiple Access (TDMA) schedule (Heinzelman, el al. 2000). Also CH aggregates data from other cluster members and send only the needed data to the sink. The main features of LEACH protocol are the large number of homogeneous and cheap nodes that can be used to monitor the environment (Szalapski and Madria, 2011). Moreover LEACH achieves power saving by aggregating collected data before sending which removes redundant data. However there is great load on the cluster heads, new nodes are selected to be cluster heads after duration of each round.

Threshold-Sensitive Energy Efficient Sensor Network Protocol (TEEN) is another example on hierarchical routing protocols (Manjeshwar and Agrawal, 2001). This protocol aims to reduce the number of message that are sent on the network by accumulating data until it reaches some predefined threshold (Hard Threshold). If the data reached the hard threshold it will not be sent unless the difference between the source data and destination data reaches other threshold (Soft Threshold). Data goes through the network using the cluster head of each sub-tree in its way.

Hierarchical-based networks main drawback is that it could not be used for wide regions. The construction of the network and the dynamic clustering brings extra overhead to the network. Moreover if nodes memorize the route nodes to the sink this would increase the memory usage in addition to computations and route calculations which consumes node's energy. This study reviews sample energy aware protocols, these protocols could be hierarchical based, tree based or cluster based protocols for WSNs. The researchers who depend on hierarchical based topologies try to benefit from the network levels in extending the range of the WSN and at the same time use the large number of sensor in the network to prolong the WSN lifetime.

2.4 Load Balancing Techniques in WSNs

Most of the time it is assumed that all nodes in WSNs are homogeneous, but actually homogeneous networks are rarely exist (Wajgi and Thakur, 2012, a). Heterogeneous networks are used most of the time in most of the applications. There are three types of resource heterogeneity in WSNs; computational heterogeneity, link heterogeneity and energy heterogeneity.

Computational heterogeneity means that some nodes have more powerful microprocessor or memory than other nodes which increases their ability to do more complicate processing. Link heterogeneity means that some nodes have greater bandwidth, or longer transmission or receiving range than normal nodes which means more transmission reliability. Energy heterogeneity means that some nodes are line powered or replaceable battery. The energy heterogeneity is the most important one to prolong the network time, because the other two type lead to consume more energy (Wajgi and Thakur, 2012, a).

To prolong the lifetime of WSN we need to save the nodes energy to the longest possible period of time. Many techniques are used to save node's energy. Sensor node consumes battery in sensing data, sending and receiving data and processing data (Babar and Halabi, 2010). It was found that transmission of messages consumes most of the node energy; hence there exist many studies focus on finding new techniques to route data with the minimum possible effort to save node energy.

According to the survey in (Wajgi and Thakur, 2012, a) 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. Their way of understanding the load balancing in WSNs leads the work on the technique. In this part of our research some techniques that try to achieve load balancing in WSN are discussed.

2.4.1 Achieve Load Balancing by Constructing a Balanced Tree

According to (Chung and Jiang, 2011) the load balancing in WSN is achieved by constructing the network in a balanced 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 the paper is based on AODV-Shortest path Algorithm and BLLCT algorithm. 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 loaded nodes that have many children should have more freedom in choosing the candidate parent.

BLLCT depends on constructing the tree from the beginning to be load balanced. It deploys the tree on two phases; first from top to bottom to select children and parents, then from bottom to top to choose the parent that is less loaded from a set of candidate parents resulted from the first algorithm.

The technique in (Behzadan, et al., 2011) also focuses on constructing the WSN to find a load balancing algorithm. The tree construction takes place on two phases, the first one until reaching the bottom as the tree construction starts from the sink on the top. The nodes in the bottom have the right to choose their parents according to their energy (best bid). The child requests to join a parent and all candidate nodes send their bids (offers) then the child picks one of them to be its parent. The child tells the chosen parent that it was selected, and informs that parent about the maximum and minimum bids it got such that the new parent will get some information about nodes in its tier (level). Node could be selfish sometimes according to game theory. Child node receives many bids from potential parents, it has the ability to choose any of these parents but because it prefers the self-interest it would choose the parent that provides more stability and energy with minimum number of children for the long run. In this protocol nodes can play the role of child and it could play the parent role at the same time.

In (Huang, et al., 20110) a tree based routing protocol is presented with some additional ideas to apply load balancing and prolong the WSN's lifetime. The number of neighbors for any node in the tree should not exceed a predefined limit; which help in reduce the load on network nodes. When some data is received, node aggregates some messages to avoid the large size of packets. This aggregation takes place using some aggregation algorithm. The network lifetime is calculated by determining when the first node in the network dies. Also each node in the tree should have a parent. Each parent node has many children that are connected to it.

It is believed that the energy consumption comes from the rate of sent packets between each parent and child and the number of children for non-leaf nodes (Huang, et al., 20110). He is trying to work on these two points to achieve load balancing. Nodes in this protocol are allowed to select their parents according to some conditions and rules. The main three rules in this protocol are: (1) few children first; mean that node chooses its parent by selecting the candidate with fewer children number. (2) Few neighbors first: to select candidate parent with few neighbors. (3) Short network distance first: the node with shorter distance has higher priority to be chosen as parent. When a node wants to choose its parent, it calculates the weight of the candidate parents using the number of children, neighbors and distance from sink (node level), then chooses the parent with the highest weight. When a node wants to join the tree it waits for a REQ message from the candidate parents, and once receive the message it sends an ACK message to the chosen parent to tell that it accept being its child. The parent broadcasts NEWS message to all it children to tell that the number of children was increased by one. In a different case, the node could receive a REQ message while it has already a parent, but this invitation could lead this node to compare the current parent with the new competitor node, and it makes the comparison and can change the parent. If the node chooses the new parent it informs the new parent and the old one as well that it is changing the parent. And the parents broadcast news message again to inform neighbors with the new situation. When a node gets a new parent it broadcasts REQ message to inform its neighbors that it is ready to receive new children. And this process keeps until reaching all the tree nodes.

It is very helpful to build a balanced topology from beginning to achieve load balancing in the WSN. For sure this could help in prolonging the lifetime of WSN. For that goal, protocols which focus on constructing a balanced topology try to create multi paths from each node to the sink, such that the alternative paths could be used later on. This kind of techniques assume that nodes are all homogeneous and they all have the same activity in sensing and transferring data, as mentioned at the beginning of this section. Practically different sensor activities lead to find different loads on the network after start transferring data. So the network will not remain load balanced for a long time.

2.4.2 Achieve Load Balancing by Calculating Next Hop Weight

In (Xue, et al., 2011) the protocol classifies nodes into two categories; normal nodes and sink nodes. Normal nodes are static nodes, which have the same radius, transmission range, link capacity, energy and all other properties. Sink nodes are not limited by energy; they also could be link powered or have rechargeable battery. It is assumed that the lifetime of the network is when one node is completely consumed. Level of node is called node-grade, means that sink is the highest grade, and the children of sink 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 reaching the sink. Choosing the next hop depends on the weight of that link, the weight is calculated using the distance (grade) of node, and 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. This is discussed in more details in chapter 4.

In this algorithm, each time a packet is going to be sent from a node there should be a loop that has a complex equation on all candidate nodes to find the best next hop. The disadvantage of this algorithm is the complex processing with every message being send; which means more energy consuming in computations and more delay. Another important thing that was not taken into account in this algorithm is the load balancing on the whole route. Selecting the next best hop does not mean that the entire path is good for routing, we can find a node full of energy but its next hop leads to a bad route.

2.4.3 Achieve Load Balancing by Data Compression

In (Cao and Yu, 2011) a different technique is provided. The idea is to distinguish data into two types: normal data, and abnormal data. The normal data is compressed and being sent normally as a vector of data. This victor is reconstructed at the sink and a projection process is achieved to data. Abnormal data which is send immediately without any delay without considering energy issue. By compressing and aggregating data from different network locations this algorithm can provide load balancing in the network.

Indeed compressing data reduces the sent data through the network. Redundant data could be reduced too. Also data projection at the sink maybe reduces the computation at other nodes. But actually depending on this technique only is not enough to achieve load balancing and prolong the lifetime of WSN.

2.4.4 Load Balancing Techniques Inspired from Animals' Behavior

Studying the behavior of animals encouraged some scientist to inspire some ideas from these animals such as ants, bees and birds. In (Almshreqi, et al., 2012) a new protocol was inspired from ants' behavior as the ants used to find better routes and overcome obstacles, it was called SensorAnt. The new algorithm is based-on ant colony optimization. It was designed to give better performance and more energy efficiency. The presented protocol uses on-demand and proactive mechanisms for routing. The on-demand is used to find new routes to the sink. On contrast the proactive mechanism is used while routing data to save data about the nodes in the tree while routing data.

SensorAnt has two types of ants used in this protocol: Ant-Forward (Ant-F) and Ant-Backward (Ant-B). Each node maintains a routing table which contains information about available routes to the sink. When some node senses data and want to send it to the sink, first it checks if there is an existing route in the routing table. If not, it broadcasts Ant-F to start seeking for a new route. To find a new path an Ant-F is broadcasted from the source node, and each node in the way keep broadcasting this packet until finding the way to the sink. The path that has the highest value of minimum residual energy and average route energy will have the highest chance to be chosen as the route path. Node could receive the same Ant-F many times, and in this case it will drop the redundant received packets. Path recovery is the process of finding new routes and update routing tables at each node. Path recovery process takes place periodically by broadcasting Ant-F to maintain the network and try to find better routes to the sink. This process is used to achieve proactive mechanism in this protocol.

The disadvantage of this routing protocol is that the routing table contains large data especially if we have large number of sensors in the network. If the node is far away from the sink data will be larger, because node should save some pieces of information about network nodes in its way to the sink. Another disadvantage of this protocol is that if there is an existing route from the source to destination in the routing table the node will use it to send data, and will not try to find another one to achieve load balancing. Finding new routes is take place periodically which means more overload and consuming more energy even if the whole process is greatly needed or not. Though there is no energy consumption to construct the network topology at the beginning, while considering on-demand mechanism to find routes, but when trying to find a new route more energy is consumed. Broadcasting Ant-F from all nodes to create routing tables makes a storm of packets in the network; this could be a big disadvantage of this protocol.

Periodic route recovery in SensorAnt could create a problem not make a good solution; because it broadcasts packets as a storm periodically even if not needed, and this increases the number of control packets in the network and may cause of many collisions and redundant packets to be sent though the network.

2.4.5 Cluster Based Techniques to Achieve Load Balancing

Clustering technique in (Waigi and Thakur, 2012, b) is trying to find a new way to save the network energy and prolong the WSN lifetime. In this technique the network is divided into groups of several nodes, each group is called cluster, and it has a cluster head (CH) and other normal nodes. The cluster head is connected to all nodes in the cluster while other nodes in the cluster should see only the cluster head. The cluster head should use all its transmission power to see all cluster nodes, while normal nodes can adjust their transmission power to reach the CH only; this could help in saving a lot of node's energy. Also some nodes in the cluster could be sent into sleep mode. When the node is in sleep mode only the minimum functionalities are still working to let the current cluster head weak it up. This protocol is helpful when there is a high density of nodes such that many nodes are very close together. The sleeping nodes would be used later on to succeed the current CH.

The CH has the role of saving the energy of the cluster and the network. CH also aggregates data from nodes in its cluster and remove redundant information to reduce the size of data that is sent from the cluster. There should be also a periodic check for the CH energy. If the energy of CH reaches some dangerous level then it should be changed with a new node that has more energy. Here comes the turn of sleeping nodes. CH asks one of the close sleeping nodes to wake up and succeed the CH is controlling the cluster. The old CH becomes a normal node, while the new CH should connect to all cluster nodes and tell them that it becomes their new CH by sending Hello message. Also some nodes will not be reached by the new CH and these nodes should try to connect to another cluster.

Although the clustering technique is used to increase the network scalability but there is still some weaknesses in this protocol. To let this technique work efficiently there should be high density of nodes to guarantee the existence of sleeping nodes that will handle the role of CH later on. Actually even with the existence of the alternative nodes of CH in the first round, but handling the role of CH could be a problem later on because there is could be no alternative nodes to handle this role after several rounds.

It is true that a lot of nodes energy could be saved by this technique but actually the load on CH is very high which leads to consume CH very early. Another disadvantage is the nodes that are far away from the new CH could be lost. The problem of the high load on CH is become greater with those clusters which are close to the sink, as the

CH at these clusters has to handle some load from all other clusters, this accelerates the death of the CH at these clusters.

2.4.6 Location Based Techniques to Achieve Load Balancing

A challenge that has been mentioned before is the unmanaged distribution of sensors in some area. Spreading the sensors from a plane for example with the existence of some obstacles on the area such as some mountain can find some area that is empty of sensors and cause some holes in the network. For that reason (Le, et al., 2012) tries to find a new scheme for load balancing in WSN, while fixing the expected holes (not covered area) in the network. There are some assumptions in this protocol such as nodes are supposed to know previously their positions, even by using GPS or other available positioning service. Source node also knows the position of destination node. All nodes have the same radio range and share all other properties, in other words all nodes are homogeneous. It is also assumed that the energy consumed by computations is very small comparing to transmission cost.

The protocol is divided into two phases; Network construction and data routing. Network construction contains (a) knowing more information about the nodes (b) collecting more information about the existed holes.

Holes in the network are drawn as polygons, and disseminated to all nodes. Nodes will have information about the place of the hole. This may help in finding uncovered areas or disaster areas. If some node wants to send data to a target destination while there is a hole in the way between the source and destination, the packets are sent to some anchor point in the way then to the destination as this is the better way to go around the holes. The network construction is basically lies in finding the holes in the network and broadcast it to nodes. On the other hand the data routing lies in finding the best way to the sink while taking care of the holes in the way. The load balancing is achieved when the source try to find the best route to the destination while going around the hole in the network.

This protocol has some disadvantages in my opinion; computing and approximating network holes consumes a lot of node's energy. Storing the information about these holes is consuming node's memory as well. If we have the position for each node, and the position of expected holes and obstacles then it should be easier to find a routing algorithm that guarantee the best path and shortest way from source to destination. Controlling this should be done in the sink or controlling unit that has powerful resources not in the sensor nodes. Actually this protocol is seems to be a very expensive protocol to be used for WSN because of some reasons; Finding the position of each node is expensive even any technique was used (GPS or any positioning services) as all of these techniques consumes a lot of energy. Another thing is the large data that is going to be processed to find the network holes. If the hole is very large, or if there are many holes in the graph then there is a large data to be processed, disseminated, and then stored in node's memory. All of these things, in addition to fetching and processing the stored data every time trying to find the best path consume the energy of WSN nodes.

CHAPTER 3

Dynamic Load Balancing Protocol for WSNs

3.1 Overview

Considering the challenges in WSN, mainly the limited resources of sensor nodes in addition to the distribution of nodes, this research presents a protocol that would increase the lifetime of WSN. The main properties of our technique are the following:

- Consider the load balancing in network construction at the first phase. If the network is well balanced from the beginning this would help to enhance the routing phase as shown shortly in the next sections.
- Consider the load balancing while routing data, taking into account the game theory rules.
- Use the minimum possible control messages.
- Use data filtration to remove redundant packets and save more energy.
- Keep the simplicity in the presented algorithms and remove complex calculations.
- Reduce the delay when sending or analyzing data packets.
- Reduce the memory usage to the minimum possible level.
- Use routing paths through neighbor nodes to best utilize the network resources.
- Benefit from any additional number of nodes in the network and exploit the new nodes to increase the performance and scalability of the network, which leads to prolong the network lifetime.

DLBP presented in this thesis aims to prolong the WSN lifetime. DLBP takes into considerations the network 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. DLBP main idea is inspired from Game theory. The way used by DLBP is to adjust the used paths dynamically while routing data. In Game Theory, the nodes have a strategy as a team, but sometimes a node could be selfish to save its energy.

DLBP includes three main phases:

- Tree construction phase.
- Data filtration phase.
- Dynamic load balancing phase.

3.2 Tree Construction Phase

Minimizing the control messages to the maximum possible limit is the main goal when constructing the tree topology. Some enhancements on (Almomani, et al., 2011, a) were achieved to give better performance and remove the not needed messages while constructing the tree. Only the tree construction phase with some updates is used in DLBP as will be shown shortly. Tree construction starts from top to bottom. The physical ID of the node will be used at the beginning until the node gets a logical ID. Logical node ID contains pieces of information such as the level of the node, the parent, and also a full route to the sink. Each node in the tree is a child of another node. The child node ID is taken from the parent node ID concatenated with the child's number as given from the parent. The number of children for any parent is limited to some number to give other nodes in the same level better chance to get children. As an example of generating the ID of a child, let's make an assumption that the number of maximum allowed children for any node is ten, and each new hop will be represented by one digit, hence every new child will extend the parent ID by one digit. As in example, assume the parent ID is 101, and it has two children; 1 and 2, the first child ID becomes 1011, and the second child ID becomes 1012. This ID tells us that the new child (1011) is in the forth level and there are three hops to reach the sink because each digit in the ID represents a level on the tree. Note that the sink node is considered to be the first level.

Participation in the routing process requires each node to have a logical ID. Once the node gets a new logical ID from its parent it becomes ready and it informs the neighbor nodes that it is ready to have some children and route data. DLBP chooses the tree construction from (Almomani, et al., 2011, a) because it is a tested protocol that gave good energy saving. DLBP has made a further enhancement to this tree construction approach mainly in these points:

- DLBP has removed some control messages. The functionality of these messages is achieved using other existing messages that already contain the same piece of information. Enhancing the header of control messages by adding small fields could replace other messages. In this way we reduce the control messages thus reduce the routing overhead. The removed messages are the following:
    - Unready message.
    - New Node message.
    - Request Parent message.
    - Change ID message.

  - Inform message.

- Cost of link is a value that is assigned to each link. Initially it is initialized to one. The cost is a logical value that is used to compare which is better to choose from two paths. It does not contain a real value related to the consumed energy or distance between nodes but it is used to help the node to choose between two paths. The cost will be discussed in more details in next section when talking about the routing phase.

Table 1 shows the control messages of DLBP that are used to construct the tree with the functionality and main attributes of each message:

Table 1: DLBP tree construction control messages

Message	When to be sent	Actions by receivers	Main attributes
Ready	When a node gets a logical ID and wants to inform neighbor nodes that it can accept new children.	1. If the receiver node does not have a logical ID it would add the sender information on waiting list until choosing a new parent. 2. If the receiver node already has a logical ID it will add the new node to the neighbors or brothers table according to the sender's ID.	1. Sender ID 2. Node Energy 3. Number of existed children
Engagement	This message is sent from a node that still without a logical ID to request a parent. This message is sent to the chosen node only to ask it for being its parent.	If the receiver node still has a place for a new child it will reply with Engagement Acceptance message.	1. Node physical ID (Mac) 2. Node energy. 3. Distance node ID
Engagement-Acceptance	This message is sent from parent to inform a node that it has been accepted as a child. And provide the node with its new logical ID	1. The receiver node will set its logical ID to the received one. 2. The link cost between the child node and its parent is initialized. 3. Broadcast a ready message.	1. New child logical ID. 2. Node energy 3. Distance node physical ID

When a node receives ready messages from neighbor nodes, it holds all messages for a while in some buffer for a short period until receiving other ready messages with new bids. The node chooses the best offer from the available list according to some metrics such as node energy, node level and cost. The node send engagement request to the candidate parent. If the candidate parent accepts the node to be its child it replies with an engagement acceptance message. If the node does not receive and engagement acceptance message for a pre-defined period of time then it chooses another candidate parent and sends it a new engagement message. The waiting period of time is previously defined as a simulation parameter. Once the node is accepted to be a child for some parent it initializes the cost and sends ready message to neighbor nodes. Also the nodes that have sent their bids before will be added to the neighbors table. A limited number of children are allowed for each parent node to give other nodes in the neighborhood a better chance to have children.

Control messages are used to construct the tree and to build some routing tables in the nodes. Each node will get some pieces of information about the neighbor nodes and save them. The expected scenario for tree construction takes place when the sink broadcasts a ready message to the nodes around. Definitely at that moment nodes still do not have logical IDs so they will reply with engagement messages to the sink. Once the sink get these engagement messages it would reply to some of these nodes with acceptance messages without exceeding the allowed number of children. Once any of these nodes get a logical ID it would broadcast ready message in its turn. The process continues until reaching the leaf nodes. The first note here is that receiving ready messages from neighbor nodes would be exploited to build neighbors table. Each node receives ready messages will keep them until choosing its parent. One of the senders would be chosen as a parent and others will be added to the neighbors table.

Tree construction process allows the tree nodes to get logical IDs, have parents, and build neighbor tables. The collected information about neighbors will be used while routing the data to find alternative paths to the sink.

3.3 Data Filtration Phase

Main functionality of the sensor node is sensing the environment and sending data to the sink through the best available route. 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. Minimizing the sent data could effectively prolong the WSN lifetime, as transferring data consumes most of node's energy. DLBP filters data that come from different children and drop redundant messages. The data that come from children nodes of same parent could be compared together to find 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 the same period of time and then redundant messages should be removed.

Indeed it would be really hard and illogical to filter all data that comes through some node. Therefore the main restriction on that is filtering and aggregating data that comes from children only. Filtering children data is acceptable because of two reasons; first the historical records that should be used in comparison process will be smaller. The other thing is that the probability of two neighbor nodes to sense the same event is very high. At the same time, it does not make sense to compare data that come from two nodes if the distance between them is large even though the sensed data was the same. Using information about data packet's first source, DLBP can find whether the received packets are from neighbor nodes or not. The actual data filtering takes place in 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 that takes place separately. It is actually part of the routing data phase as will be explained in section 3.4.

3.4 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 is trying to provide a technique that keeps the load balancing of the WSN even after sensors start collecting data from the environment and send it 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. Other routing protocols that consider dynamic load balancing allow the node to make complex calculation each time it sends a packet to find the best next hop. The presented technique considers the load balancing at all phases of the WSN including tree construction as shown in section 3.1, and during data routing. On the other hand, DLBP tries to remove complex calculations and prevent delaying messages before being sent.

Game theory is a decision making theory that consists of three components: a set of players, a set of actions that could be taken by any player in the game and a set of strategies. The strategy is a set of actions that are taken by each player in the game (Machado and Tekinay, 2008). It is some kind of plan that tells the player which action to take at a certain time. The utility represents the goal of player. Each player in the game can act selfishly and increase its gains; in this case the player wants to increase its own utility (Srivastava, et al., 2005). Also the player can cooperate with other game players to achieve the best utility. Utility function assigns every output from the game to some value. The higher value is the best for the player (Machado and Tekinay, 2008). There is a strategy for the whole game while each separate player in the game has its own strategy which is a set of actions to be taken when the game is actually starts. The utility function is the function that shows the preferences of each player in the game (Srivastava, et al., 2005). Nash equilibrium is a set of actions taken by the game players such that any other set of actions will not come out with better utilization (Zhang, et al., 2008).

Nodes in the game theory 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 tries to find the best equilibrium 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 this sink and help in achieving load balancing on the network.

To understand the use of Game Theory in our research let us take a football team as an example. The football team consists of eleven players, and each player has a role. Sometimes a player could be selfish and keep the ball with himself without passing it to other players who have better positions. Other times the player has to be selfish, such as the striker in front of the goal, he should be selfish and think of shooting the ball into the goal instead of passing it to other team player. Being selfish is bad sometimes and is a must in different situations. To be a member of the team, there should be some rules to organize when to be selfish or not. In WSN the same rules should control the nodes decision about passing every data packet to the next hop, or drop it according to the energy level, or filter it and drop redundant packets if needed.

DLBP in the first phase of tree construction tries 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 and it would be increased once that is needed. The cost of routing data from parent node to the sink is accumulated to the cost between the child node and parent. If the node has many options it will take the best option and use it unless a change on its cost take place. Figure 6 shows the cost calculation in DLBP in a simple way. The Initial cost of each link is set to one. 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. Note in Figure 6 that node D has two paths to the sink; one through node B which cost is 2, and another one through node C which cost is 3. Node D chooses node B to be the default next hop until a new change take place on the whole path which forces node D to make a new check and find the best path again.

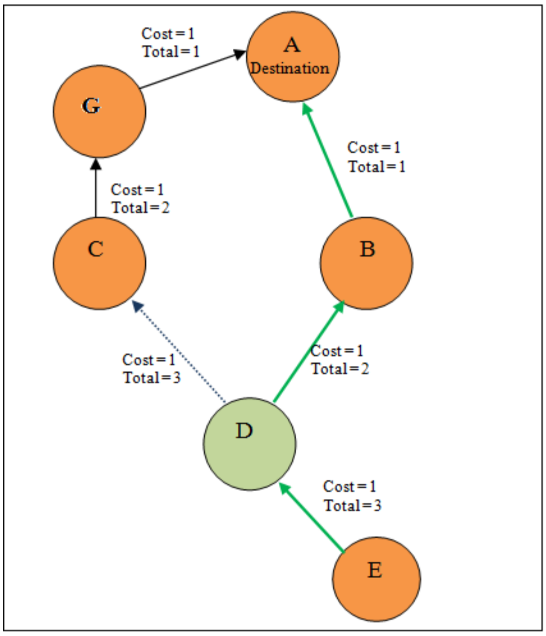

Figure 6: DLBP accumulative link cost.

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 try to find a list of alternative paths with less cost and use them. Figure 7 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 Figure 7 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 Figure 7 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 from these two options for now. In figures 7 C and D another changes take place on the cost, therefore, routing path alternatives are keep changed dynamically.

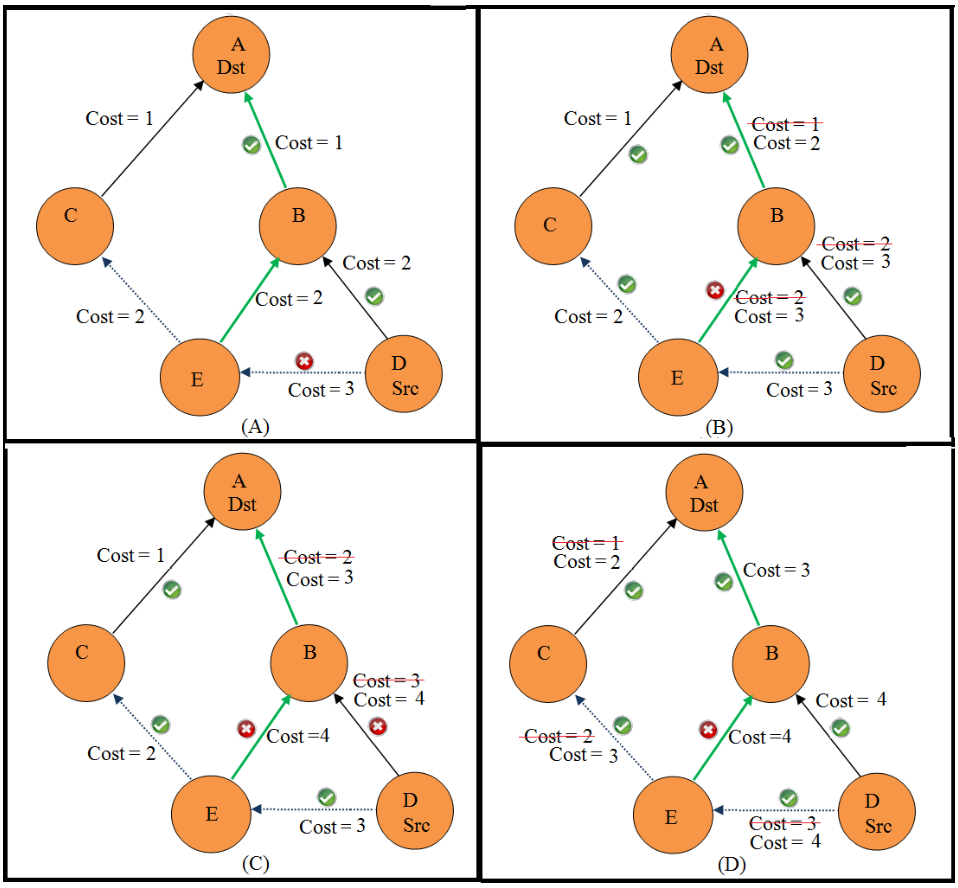

Figure 7: Example of path choosing based on links cost. The next hop is the link with minimum cost. Note that ☑ represents the paths with lowest cost, and ☒ represents the paths with higher costs that will not be selected unless there is no other option. Note the ☑ and ☒ after each cost change.

The load balancing techniques 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 regarding 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 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 some branch of the tree makes a high load on its parent, then the parent node should inform that branch to reduce its messages to be in the normal range. Controlling sub branches is done by controlling the head of that sub branch. To control the receiving amount of data that should be received from a node, the following points should be considered:

- Cost of links between nodes is the value that forces the node to find the path with the lower cost. DLBP inspires this idea from the water flow. When the water flows on some area we can note that it takes the lower lands to be the waterways. DLBP directs the load from the overloaded paths by increasing their costs and consequently direct data packets to find lower cost paths.

- DLBP replaced control packets by adding small values to the data message header. In DLBP there are no control packets to be sent during the data routing phase. If the cost of some link was changed, then neighbor nodes should be informed about that change. Usually, routing algorithms send a control packet to inform neighbor nodes that the link cost has been changed. 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 add to the header of data packets, and nodes that receive that data packet will check for the header and update the cost if needed. The good thing here is that DLBP eliminated a control packet from being sent.

- DLBP uses data unicast option to send data packets from any node to another. Since we are dealing with wireless nodes, then the nodes in the neighborhood area are already listening to every message and reading the header to find out whether this packet belongs to them or not. In addition to this, in DLBP the node will read the packet header to update its information about the sender node. Each node should update the header of data packet before forwarding it to the next hop.

- If a node receives a packet from one of its children or neighbors then it will update the counter of received packets from that node. Next a simple calculation is achieved to find the ratio of received packets from that node to the total received packets as shown in the equation below:

$$\text{Received packets ratio} = \frac{\textit{Received packets from node}}{\textit{Total received packets}}$$

The ratio should not exceed some threshold limit as will be discussed shortly. But if that limit was exceeded then parent node has to alert the node that exceeded the limit to increase the link cost between the two nodes. This is achieved by adding the node ID that is needed to be alerted to the data packet header. If the node did not exceed its messages limit then that header will remain null. If a node receives a data packet and finds its ID in the toBeAlerted value, this means that it has to increase the link cost.

- After many changes in the paths the network will reach a stability situation. This means that a slow adjustment process is taking place on the network to make it well-balanced during sending data packets.

The question here is how DLBP would balance the load dynamically on the WSN. The answer is simple, 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. The other thing is the recalculation to find alternative paths will not take place when sending any new data message. On the other hand, to reduce the routing overhead, DLBP extends the header of data messages to be used instead of sending control messages. The header size of data messages could be increased by some bytes but even with this increase it still better than sending full message to inform the neighbor nodes with a small piece of information. 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. Table 2 shows the extra fields that are added to data packet and used by DLBP. Note that the size of the needed headers is 8 bytes, which is a small value.

Table 2: Data packet header

Header Value	Size	Description
Cost	4 bytes	Best cost of sender node that is used to send data to next hop
toBeAlerted	4 bytes	Physical ID of node that exceed sending limit and should be alerted. By default this value is set to NULL
Total Size	**8 bytes**	

Figure 8 shows the algorithm of DLBP. Note that the receiver always read the header of the received packet and update the information about the sender in the previous hop. If receiver node is the destination and the packet is not redundant it will be sent to application module to read and analyze data. If the receiver node is just the next hop of the sender node it will filter the message to make sure that it is not redundant packet then it will forward the packet to the next hop. In all cases, if the receiver node is the destination of the packet then it will update the counter of received packets from the sender node. This counter resides in neighbors table, it counts all packets received from some node. Note that "To Application" means that the packet reached its final destination. "To MAC" process means to broadcast the packet to neighbors after defining the next hop ID in the packet's header.

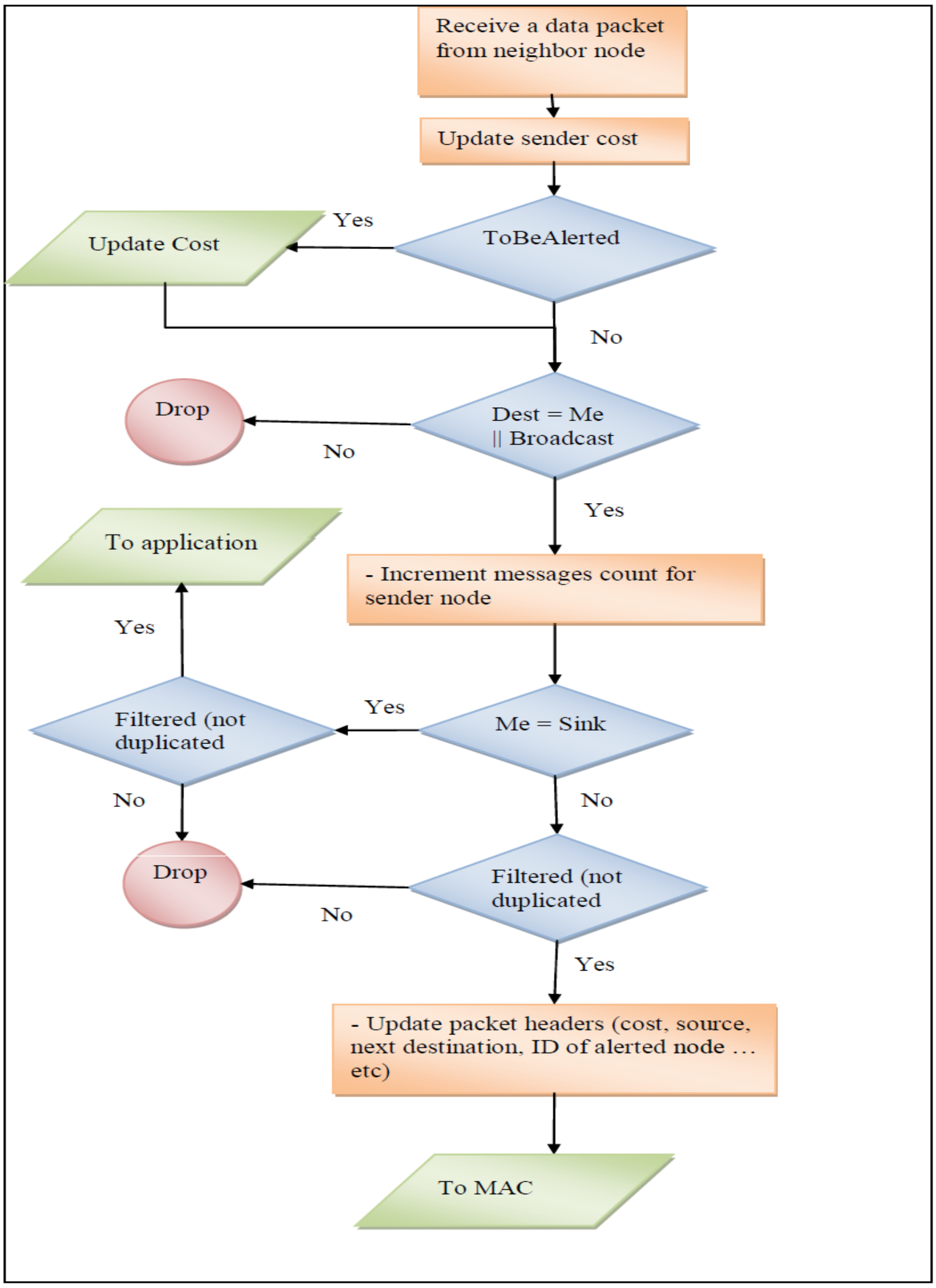

Figure 8: DLBP Algorithm

The average is calculated for neighbors and children such that each node has an equal share to send its packets through the parent node. If the node exceeded that limit with some threshold then this node should be alerted to increase the cost of its link. Adding the node ID to the header of packet means that this node is alerted and should increase the cost. To explain this in more details here is an example; Figure 9 shows node B has two children C and D. Each child has 50% of allowed messages to be sent through B. There is a threshold that should not be exceeded. In DLBP the threshold is a simulation parameter and it is initially set to 10. If any of the two children sent packets exceeded the allowed share plus the threshold (50 + 10) then parent node B should alert the node to increase its cost therefore the load from this child will be reduced.

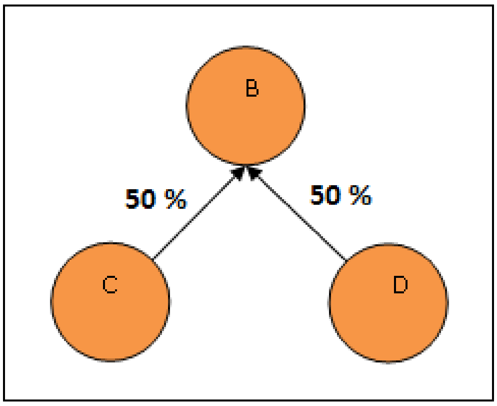

Figure 9: Children link share. Each node of children has equal share of sent messages through parent node. Nodes C and D has 50% of allowed sent packets through B.

The last thing to address is the main points that make DLBP different than other existed techniques. Here are some points:

- Acceleration: Each node has a list of next hop candidates. No need to make a complex calculation each time the node wants to send a packet. Therefore packets are not delayed anymore. This increases the routing speed and reduces the delays.

- Accuracy: Selecting neighbor nodes to be in the list of next hop candidates does not depend only on the information in the node about its neighborhood. There is a monitoring system in the network as a whole. Information about nodes and costs are going down from top levels until reaching the network leafs. This monitor helps the network to have wider and more accurate information about the best paths to be chosen.

- Simplicity: Complex calculations were removed when calculating link costs. Also running the link cost calculation process take place only if needed. After a while of running the cost calculation process, the network reaches a stability stage such that making these calculations is reduced and could reach zero.

- No routing overhead: Control messages are not used after the network gets constructed. This means minimizing the routing overhead.

- Performance: DLBP guarantees to deliver most of the sent packets to their destination. This means a high throughput and therefore a high performance.

3.5 Link Maintenance and Termination

Every routing protocol should always have alternative plans to fix errors or link failures in the network. DLBP already prepared the nodes to have 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 energy reaches a low level 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 achieved by changing to cost to a minus number instead of the current cost. When neighbor nodes find a minus 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 should be available in neighbors table. This will make the network maintenance very fast, without any additional control packets or loads.

CHAPTER 4

Results and Evaluation

In this chapter the simulation results of our presented protocol are discussed and evaluated. DLBP is compared to the protocol presented in (Xue, et al., 2011) 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 thesis. 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.
- It is one of the recent researches in this field as it was published in 2011.
- The authors of this algorithm have simulated and evaluated simulation shows good results comparing to three of widely used algorithms which are Short-Path routing algorithm (Gao, et al., 2006), STRP and the power aware version of STRP which is called STRP-PA (Ben-Can, et al. 2008)
- Xue algorithm is also easy to be implemented and the similarities between DLBP and Xue make it easier to be compared with DLBP.

Simulation scenarios took place on fixed networks with different number of nodes to compare between the two protocols in terms of some metrics, while taking the network scalability into account.

Omnet++ simulator with Castalia framework was used to implement the network protocols. Omnet++ provides an architecture that helps programmers to create their own modules using C++ language and then collect some modules to create more complex compound modules. Also the programmer can use the Omnet++ models that provide a full testing environment to evaluate the algorithms and generate outputs results. Castalia is a framework of Omnet++ simulator. It was created especially to simulate WSN and Body Area Networks (BAN). Castalia provides many models for the wireless node and implements the needed layers such as application, MAC, network and physical layers. Also the radio model and wireless channel in Castalia make the simulation results to be very close to reality as the connection between nodes takes place through the wireless channel not directly (NICTA, 2013).

Bridge test simulation was used to evaluate the two routing protocols. This application is the default implemented application in Castalia. Bridge test is an application in which sensor nodes are distributed on a bridge. Each car crosses the bridge stimulate the sensor nodes in its way to fire event to tell the sink that it has sensed the motion of that car. Increasing the number of sensors on the bridge lets more nodes to sense the movement of cars, but on the other hand the number of cars is the same. Understanding this application is very important to understand the results in the next sections.

Table 3 shows the simulation parameters that used to evaluate DLBP and compare it to Xue algorithm. The node transmission range is 50m, and this value was found experimentally. Each scenario was tested up to five times. Average results for all runs were calculated and used to draw figures. The simulation results will focus on five metrics: network scalability, network throughput, routing overhead, routing computation times, and network lifetime.

Table 3: Simulation Parameters

Parameter	**Value**
Omnet++ version	4.2.2
Castalia Framework version	3.2
Number of Nodes	50 – 300
Environment Size	50×50 – 1000×1000
Nodes mobility	Static
Application module	Bridge Test
Simulation time	15000 sec
Physical Module	CarsPhysicalProcess
Physical car intervals	5
Radio Model	CC2024
Radio TX Level	-5 dBm
Transmission range	50 m
Number of receivers (Sink)	One
Processor	2.4 Quad Core
RAM	6 GB

4.1 Network Scalability

The consumed and remaining energy were calculated on different network sizes to evaluate the network scalability. Figure 10 shows the Consumed Energy after the simulation is completed. The x-axis represents the number of nodes and shows the different scenarios that are used in our simulation. The y-axis represents the consumed energy in Joule unit. Figure 10 shows that in DLBP the nodes consume the same amount of energy even after increasing the number of nodes. On the other hand, when using Xue nodes consume 20% more energy when the number of nodes is increased. This gives us an indicator that DLBP is more scalable. Figure 11 as well shows the Remaining Energy Percentage after the simulation is completed. It is found that nodes in DLBP keep saving the same level of energy even when the number of nodes increases.

In DLBP the consumed energy remain the same even after increasing the number of nodes. The reason behind that refers to the nature of experiment and the used application in the simulation. In some applications nodes sense the environment periodically and send periodic packets to the sink with the sensed data. In our simulation the bridge test application has been used as mentioned before. The bridge test application uses the same number of cars to cross the bridge in the desired time, this explains the stable results of consumed energy amount even after increased the number of nodes.

Moreover, in DLBP if a moving car stimulate some neighbor nodes and all of them sense the motion and send a packet of that, then the redundant data will be dropped. As an example on how DLBP is scalable let’s assume that there are two neighbor nodes on the bridge, and a car moved near them and stimulated both of them to generate the same packet. Only one of the two messages will be sent to the sink and the other will be dropped on the next hop. If there are more close nodes generate data packets for the same event, one only of the generated packets will be sent to the sink and the rest will be dropped, which makes the network more scalable. On the other hand, Xue algorithm sends all data packets to the sink without filtering, so if there are ten neighbor nodes that sense the same event, then the number of generated packets will be ten times the number of packets if there was only one sensor node in that place.

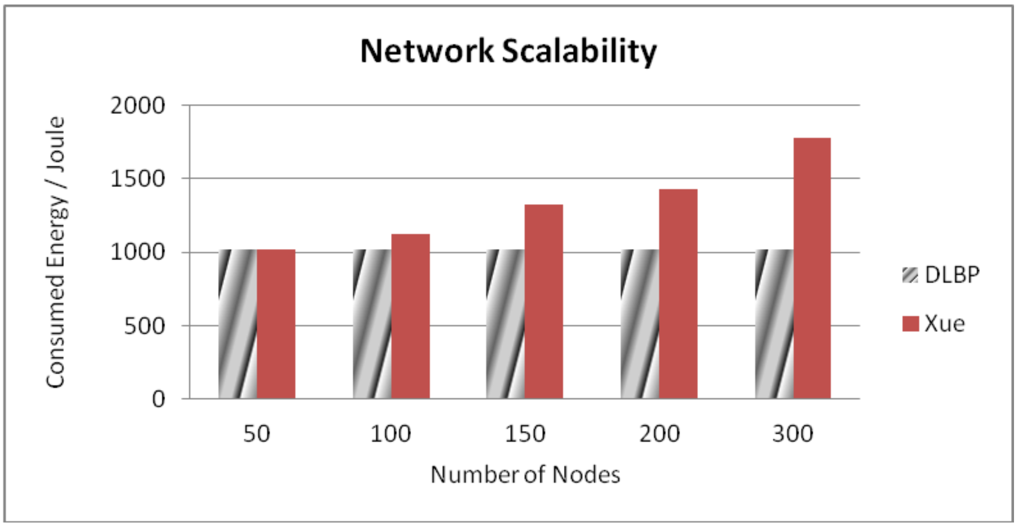

Figure 10: Network scalability by calculating consumed energy vs. Number of nodes

In Figure 11 the x-axis represents the number of nodes and the y-axis represents the remaining energy percentage. Note that DLBP keeps the remaining energy in a high level even when the number of nodes increases.

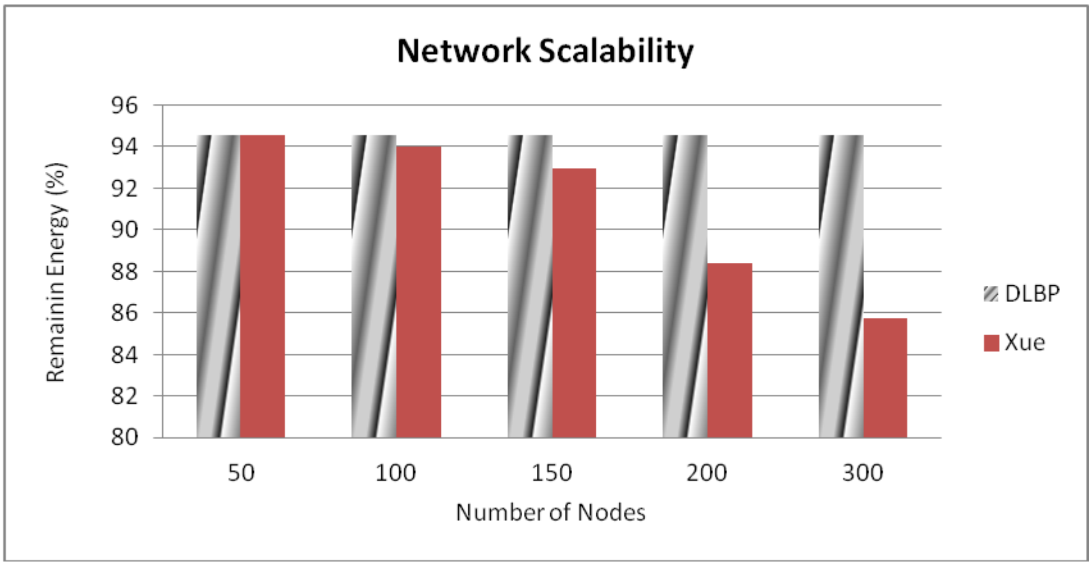

Figure 11: Network scalability by calculation nodes remaining energy percentage vs. Number of nodes

4.2 Network Throughput

The network throughput is evaluated by calculating the number of sent and received packets from each node. Figure 12 shows the number of received packets and Figure 13 shows the failed packets. When a node broadcasts a packet and neighbor nodes receive that packet, then the received packets counter is incremented by the number of nodes that successfully received that packet. If a node in the range of sender failed to receive the packet successfully, then the failed packets counter is incremented by one. The filtration process in DLBP drops redundant packets before being forwarded to reduce the redundant packets. This could be noticed from Figures 12 and 13.

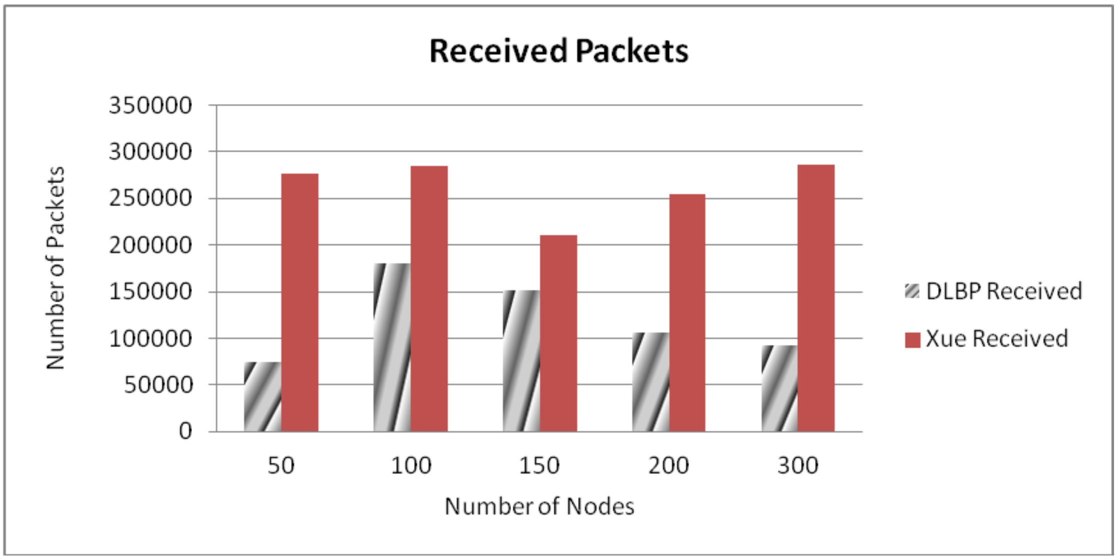

Figure 12: Number of received packets vs. Number of nodes

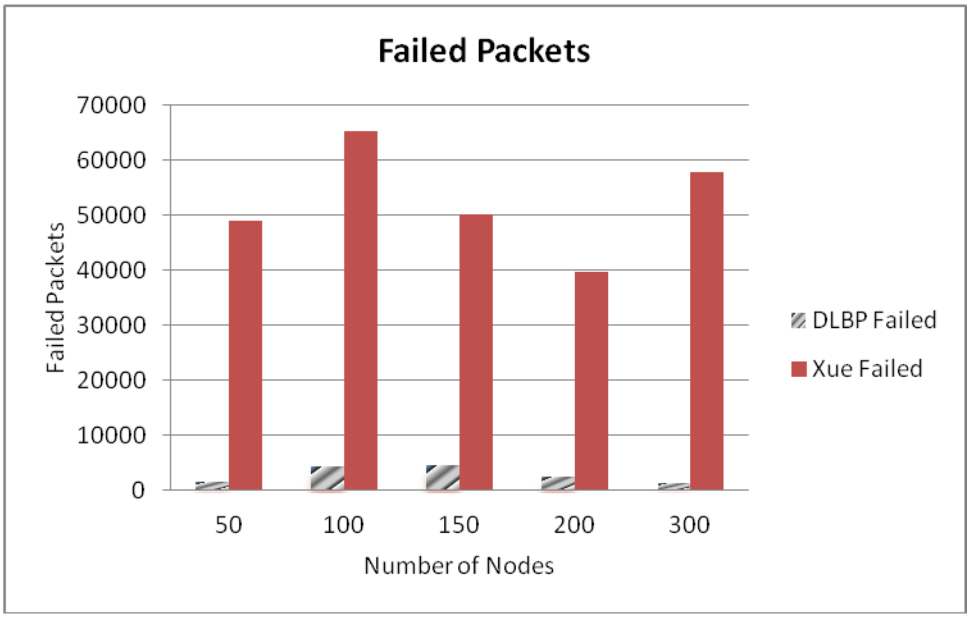

Figure 13: Number of failed packets vs. Number of nodes

The received packets ratio gives another important indicator about the ability of DLBP to deliver the sent messages. Figure 14 shows the received packets ratio, which is the percentage between the received packets to the total sent packets.

Received Packets Ratio = $\frac{Received\ Packets}{Total\ Sent\ Packets}$

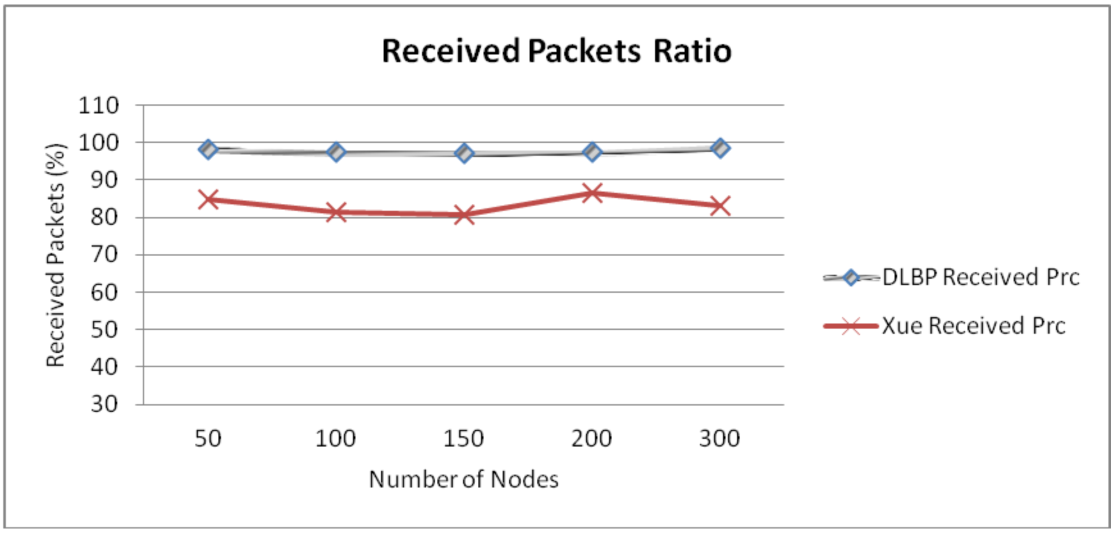

Figure 14: Received packets ratio vs. Number of nodes

On the other hand, Figure 15 represents the failed packets ratio. Note that in DLBP the failed packets ratio is very small and remains stable even after increasing the 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. Also using unicast mode to sent packets helps in minimizing the interference between nodes too. Unicast is achieved by adding the ID of the next destination node in the header of data packet. 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 throughput of the network as well. In DLBP, nodes are supposed to choose the best paths to route data packets, which means less cost and higher throughput.

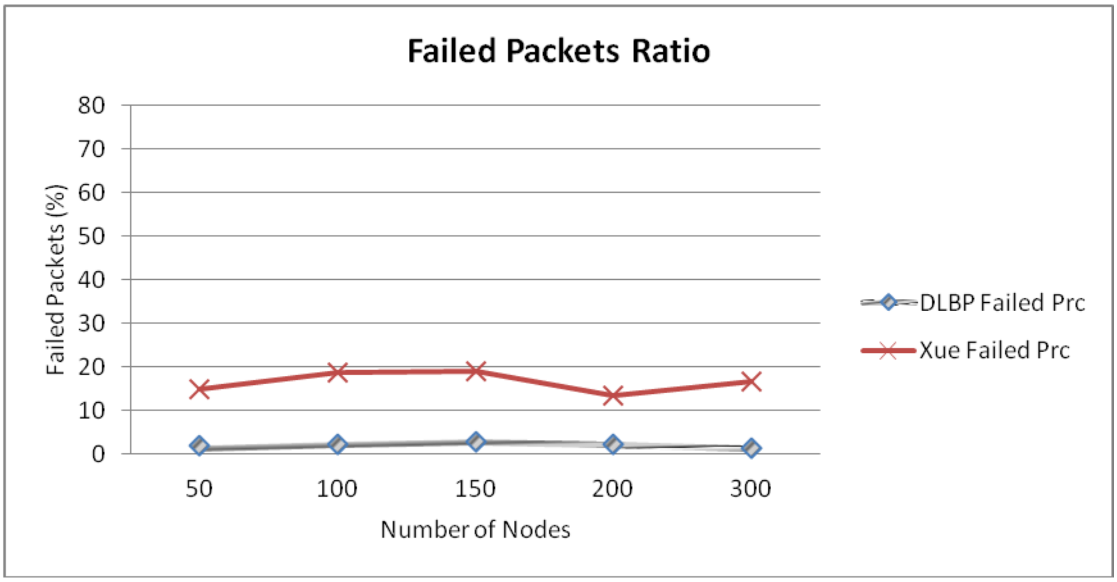

Figure 15: Failed packets ratio vs. Netwok size

4.3 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.

$$\text{Routing overhead} = \frac{\text{Total sent control packet}}{\text{Total Sent Data \& Control Packets}}$$

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.

The number of sent control packets per node is shown in table 4. It is clear from the table that the number of control packets is a very small in DLBP. The reason is the elimination of control packets after finishing the tree construction phase. On the other hand, Xue 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 data 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.

Table 4: Number of sent control packets per node

Number of nodes	**50**	**100**	**150**	**200**	**300**
DLBP	4.84	52.98	52.847	33.445	3.343
Xue	5079.536	6754.55	3654.2	4517.6	5478.65

In Figure 16 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. The other note on the simulation results is that the number and distribution of nodes plays a very important role in increasing or decreasing the number of 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.

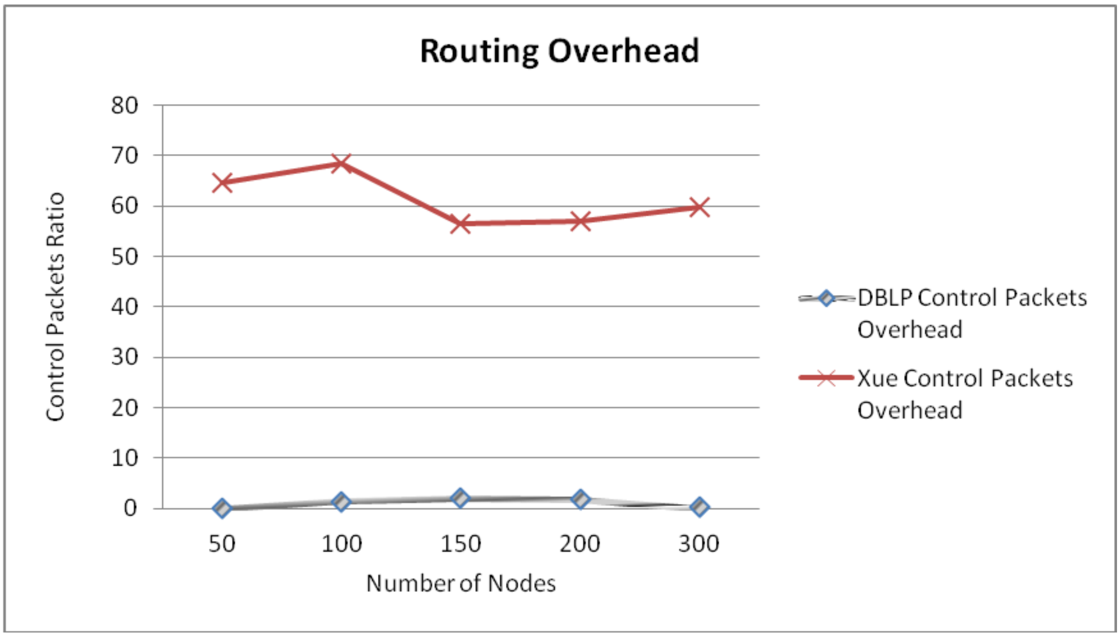

Figure 16: Routing overhead vs. Number of nodes

4.4 Routing Protocol Computations

Routing data from one node to another needs the sender node to know the next hop and define it with the sent packet. In DLBP the nodes already know the best next hop which provides the minimum cost. By default the next hop is the parent node, and then the best next hop could be changed according to data traffic. Nodes on top levels keep watching the behavior of their children and once they notice unbalanced data traffic they inform the children to do some calculations to rebalance the routing dynamically. Changing the cost of link between any two nodes directs the nodes that use this link to find a cheaper link and use it, which reduces the load on the overloaded link. After several times of adjusting the costs of links, the network will reach a stability state. If the network reach the stability state and become well balanced, then no calculations are needed anymore, which means less number of computations.

Xue protocol has a different technique. It calculates the next hop before every packet sending. To calculate the next hop in Xue algorithm, first the weight of link is calculated between the node and all of its neighbors. Then the possibility of sending through the desired link is calculated for each link. The link with the highest possibility is used as the next hop. Equation 1 shows the weight calculation. Then equation 2 shows the use of link weight to calculate the possibility.

$$w_{ij} = \frac{(E_j)^{\alpha}\, e^{-(rj\text{-}ri+1)}}{d\,(vi,\, vj)^{\beta}} \qquad \textbf{(1)}$$

$$p_{ij} = \frac{1}{n}\sum_{j=1}^{n} w_{ij} \qquad \textbf{(2)}$$

In these formulas, w_{ij} represents the link weight between nodes i and j. E_i represents the rest energy of node *i*. r_j is the grade or level of node *j*. $d(v_i, v_j)$ is the distance between nodes *i* and *j*, α and β represent the remaining energy factor and the distance factor (Xue, et al., 2011).

Table 5 shows the big difference between the two protocols regarding to the number of computation times they do. As it shows Xue do much more complex calculations than DLBP. Note that in DLBP, if the network is already well balanced after the first phase and there is just little number of overloaded links, then the number of calculations to adjust the cost will be reduced. The well distribution of nodes on the testing area plays a very important role in balancing the network. Therefore the complexity could be reduced.

Table 5: Average number of computation times for nodes in DLBP and Xue algorithm

Number of nodes	50	100	150	200	300
DLBP	2.04	64.64	57.38	55.275	3.01
Xue	2820.4	3189.55	2910	3735.06	3901.5

4.5 Network Lifetime

The goal of presenting DLBP is to increase the network lifetime. This section reviews the simulation results and how DLBP prolongs the WSN lifetime. To evaluate network lifetime a simulation scenario was tested for twenty hours on a network of 100 nodes. Figure 17 shows the network lifetime. The x-axis is the simulation hours and y-axis is the remaining energy in Joule. It can be noticed from Figure 17 that nodes in DLBP remain stable all the time, and consume energy in the same level. While in Xue protocol nodes consume energy faster after running the simulation for longer time. The reasons behind this enhancement are the following:

- Filtering redundant data reduces sent and received data packets, which saves more energy and results in prolonging network lifetime.
- Elimination of control packets while routing data reduces energy consumption and prolongs network lifetime.

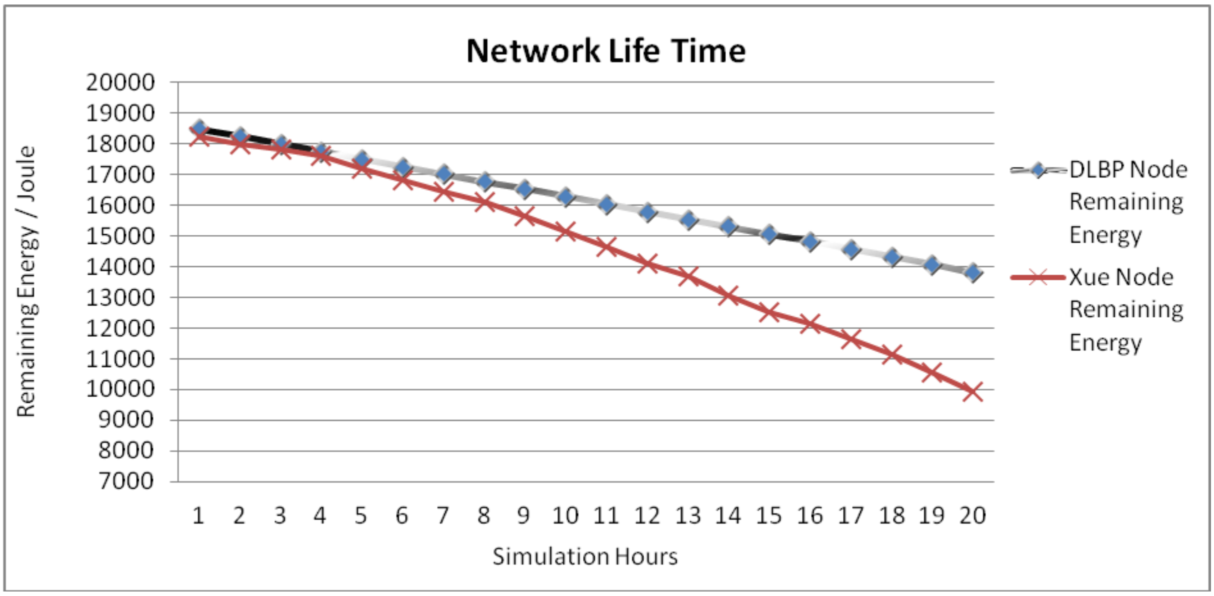

Figure 17: Network Lifetime. Node remaining energy vs. simulation hours

4.6 Summary

In this chapter we have presented and discussed the simulation results. Different metrics of WSNs have been discussed and evaluated. Different scenarios have been tested and evaluated for two different techniques. DLBP technique shows very promising results over Xue technique. DLBP succeeds in minimizing the routing overhead, the consumed energy, the total sent packets, hence DLBP succeeds to prolong the network lifetime at the end.

CHAPTER 5

Conclusions and Recommendations

This thesis has presented a dynamic load balancing protocol for WSNs based on game theory. The presented technique has been used to route data through the network while being aware about the energy consumption. The main goal behind presenting this technique is to prolong the WSN lifetime. This chapter gives a short conclusion for this thesis and summarizes the simulation results. Then some future directions and recommendations are getting suggested.

5.1 Conclusions

The presented technique succeeded to achieve a success ratio regarding the points it has tried to enhance. Below is the summary for these points.

- In phase one, the tree construction, DLBP succeeded to build a tree of nodes such that every node in the tree has a parent, and it also knows its direct path to the sink without any calculations, just using the node logical ID. Some kind of balancing was achieved on that tree to make the later balancing tasks easier and faster.

- Data filtering technique in DLBP succeeded to reduce the number of redundant data packets. Therefore, the interference has been decreased as well as the failed packets ratio has been decreased too. On the other hand, the success ratio and network throughput has been increased. Data filtering also helped to prolong the network lifetime.

- DLBP load balancing technique with the monitor of top level nodes has been succeeded to distribute the load on network nodes to prolong the WSN lifetime.

- The elimination of control packets in DLBP reduces the routing overhead. It also helps to prolong the WSN lifetime

- DLBP has been successfully reduced the calculations complexity. It has accomplished the needed tasks with few amounts of calculations that do not make a heavy load on the sensor processor or memory. The simplicity in DLBP makes it faster and lighter to be used in WSNs.

5.2 Summary for the contributions

The presented technique DLBP showed promising results comparing to Xue algorithm after running the simulation for both protocols in the same environment. Table 6 summarizes the enhancements of DLBP over Xue algorithm.

Table 6: DLBP enhancements comparing to Xue algorithm

Metric	**Scalability**	**Success ratio**	**Routing overhead**	**Processing Complexity**	**Network lifetime**
DLBP enhancement	Increased by 20%	Increased by 16%	Decreased by 72%	Decreased by 99.9993%	Increased by 20%

5.3 Future Directions and Recommendations

WSN is a hot concept in wireless communications which means that there still much researches are going on in this domain. This study has focused on some sides and techniques to prolong the lifetime of WSN, but there still many future directions that are needed to be explored in the future.

- Build a mathematical model for our technique. As we depend on simulation results only in this study we need to find the equations to find the same results mathematically.

- Consider the processing cost when finding the consumed energy. Omnet++ simulator does not take the processing of data into account which make it hard to compare between any two nodes depending on this metric.

- Consider the consumed energy resulted when receiving packets. Omnet++ simulator has a resources manager module that handles calculating consumed energy. The resources manager takes sending packets process only into considerations without considering that receiving packets consumes energy as sending packets. If the energy resulted by receiving packets is calculated this will give DLBP another step forward Xue protocol.

- Simulate the network total throughput. By considering every packet that was sent to the sink and find out how many packets reached the desired destination at the end.

- Find the processing delay.

- Consider the security issues for DLBP.

REFERENCES

Mostafa Al-Harbawi, Mohd Fadlee A. Rasid, and Nor Kamariah Noordin, "**Improved Tree Routing (ImpTR) Protocol For ZigBee Network**", International Journal of Computer Science and Network Security, vol.9, No.10 , 2009, pp.146 – 152.

Altenergymap, **http://www.altenergymag.com/emagazine.php?issue_number=08.12.01&article=links**, last visit time 5th Mar 2013.

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.

Broadband Wireless Networking Lab (BWN), **http://www.ece.gatech.edu/research/labs/bwn/corr/projectdescription.html**, last visit time 13th May 2013.

Harvard University, **http://www.eecs.harvard.edu/~mdw/course/cs263/fa04/**, last visit time 5th Mar 2013.

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.

Iman Almomani and Maha Saadeh, “**FEAR: Fuzzy-Based Energy Aware Routing Protocol for Wireless Sensor Networks**”, Int'l J. of Communications, Network and System Sciences, 2011, 4, pp. 403-415.

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.

Iman Almomani, Maha Saadeh, Mousa Al-Akhras, Hamzeh Al Jawawdeh, “**A Tree-Based Power Saving Routing Protocol for Wireless Sensor Networks**”, International Journal of Computers and Communications, University Press, Vol. 5, No. 2, 2011, pp. 84-92.

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.

Babar Nazir, Halabi hasbullah, "**Energy Balanced Clustering in Wireless Sensor Networks**", Information Technology (ITSim), International Symposium, 15-17 June 2010, Kuala Lumpur, Malaysia, pp. 569-547 .

Afshin Behzadan, Algan Anpalagan, and Bobby Ma, "**Prolonging Network Lifetime via Nodal Energy Balancing in Heterogenous Wireless Networks**", Communications (ICC), IEEE International Conference on 5-9 June 2011, Kyoto, Japan, pp. 1-5.

Guangming Cao, Fengqi Yu, "**The Analysis of Load Balance for Wireless Sensor Network using Compressive Sensing**", 14th IEEE International Conference on Computational science and Engineering, 24-26 Aug 2011, Dalian, Liaoning, pp. 100-105.

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.

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.

Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan, Massachusetts Institute of Technology Cambridge, MA 02139, "**Energy-Efficient Communication Protocol for Wireless Microsensor Networks**", Proceedings of the 33rd Hawaii International Conference on System Sciences, 4-7 Jan 2000, vol.8, p. 8020.

W. Heinzelman, J. Kulik, and H. Balakrishnan, "**Adaptive Protocols for Information Dissemination in Wireless Sensor Networks**", Proc. 5th ACM/IEEE Mobicom Conference (MobiCom '99), Seattle, WA, August, 1999. pp. 174-85.

Chung-Kuo Huang, Guey Yun Chang, and Jang-Ping Sheu, "**Load-Balanced Trees for Data Collection in Wireless Sensor Networks**", 41st International Conference on Parallel Processing Workshops, 10-13 Sep 2012, Pittsburgh, PA, pp.474-479.

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.

W. Jinghua, G. Tingting, H. Huan, and C. Yuanyuan, “**An energy and load-based routing algorithm in wireless sensor network**” in Computational Intelligence and Desig. ISCID ’09. Second International Symposium, vol. 1, 12-14 Dec 2009, Changsha, China, pp. 284–287.

N.P. Karthickraja, V.Sumathy, “**A Study of Routing Protocols and A Hybrid Routing Protocol Based on Rapid Spanning Tree and Cluster Head Routing in Wireless Sensor Networks**”, Wireless Communication and Sensor Computing, ICWCSC International Conference, 2-4 Jan 2010, Chennai, India, pp. 1-6.

J. Kulik, W. R. Heinzelman, and H. Balakrishnan, "**Negotiation-based Protocols for Disseminating Information in Wireless Sensor Networks**" Wireless Networks, Volume: 8, 2002, pp. 169-185.

Nguyen Phi Le, Nguyen Trung Hieu, Nguyen Khanh Van, "**ELBAR: Efficient Load Balanced Routing Scheme for Wireless Sensor Networks with Holes**", SoICT 2012, pp. 190-199.

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.

Arati Manjeshwar and Dharma P. Agrawal, “**TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks**”, Center for Distributed and Mobile Computing, CECS Department, University of Cincinnati, Parallel and Distributed Processing Symposium, Proceedings 15th International, 23-27 Apr 2000, San Francisco, CA, USA, pp. 2009-2015.

MCSL, **http://mcsl.ulsan.ac.kr/research01.htm,** last visit time 13th May 2013.

NICTA, **http://castalia.npc.nicta.com.au/pdfs/Castalia_AcademicPublicLicence.pdf,** last visit time 10th April 2013.

Ning Xu, “**A Survey of Sensor Network Applications**”, IEEE Communications Magazine, vol. 40, University of South Carolina, 2002.

Nurhayati, Sung Hee Choi, and Kyung Oh Lee, “**A Cluster Based Energy Efficient Location Routing Protocol in Wireless Sensor Networks**”, International Journal of Computers and Communications, vol. 5, issue 2, 2011, pp. 67-74.

B. Peng and A.H. Kemp, “**Energy-Efficient Geographic Routing in the Presence of Localization Errors**”, Computer Networks, The International Journal of Computer and Telecommunications Networking, vol. 55, issue 3, Feb 2011, Elsevier, North Holland, pp. 856–872.

Shio Kumar Singh, M P Singh and D K Singh, "**Routing Protocols in Wireless Sensor Networks – A Survey**", International Journal of Computer Science & Engineering Survey (IJCSES) vol.1, issue 2, November 2010, pp. 63-83.

Vivek Srivastava, James Neel, Allen B. Mackenzie, Rekha Menon, Luiz A. Dasilva, James E. Hickes, Jeffrey H. Reed and Robert P.Gilles, “**Using Game Theory To Analyze Wireless Ad Hoc Networks**”, IEEE Communication Surveys & Tutorials, Fourth Quarter 2005, vol. 7, issue 4, pp. 46-56.

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.

Singularityhub, **http://singularityhub.com/2010/02/24/tiny-sensor-could-run-for-years-harnessing-energy-from-its-environment/** , last visit time 5th Mar 2013.

Tommy Szalapski and Sanjay Madria, “**Energy-efficient Real-Time Data Compression in Wireless Sensor Networks**”, Mobile Data Management (MDM), 12th IEEE International Conference, vol. 1, 6-9 Jun 2011, Lulea, pp. 236-245.

Dipak Wajgi, Nileshsingh V. Thakur, “**Load Balancing Algorithms in Wireless Sensor Network : A Survey**”, IRACST – International Journal of Computer Networks and Wireless Communication, 2012.

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.

Wikipedia, **http://en.wikipedia.org/wiki/Sensor_node**, last visit time 13th May 2013.

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.

Yan Yu, Ramesh Govindan, Deborah Estrin, "**Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks**", UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023, 2001.

Lun Zhang, Yan Lu, Lan Chen and Decun Dong, "**Game Theoretical Algorithm for Coverage Optimization in Wireless Sensor Networks**", Proceedings of the World Congress on Engineering, vol. 1, London, U.K. 2-4 Jul 2008, pp. 764-769.

نظام فعال لموازنة الأحمال في شبكات الاستشعار اللاسلكية

إعداد
حمزة جميل الجواودة

المشرف
الدكتورة إيمان موسى المومني

ملخص

إن شبكات الاستعار اللاسلكية هي من أكثر المواضيع المطروقة في مجال الأبحاث والدراسات في الوقت الحاضر. الإمكانات المحدودة لدى وحدات الاستشعار اللاسلكي مثل الاعتماد على بطاريات قليلة الطاقة وغير قابلة للشحن أو التبديل، وصغر حجم الذاكرة وكذلك محدودية قوة المعالج. كل ما سبق يعتبر تحدياً كبيراً للباحثين. ومن أكثر التحديات بناء نظام موفر للطاقة يعمل على الشبكة اللاسلكية لإطالة عمرها لأطول مدة ممكنة.

هذا البحث يقترح آلية فعالة لموازنة الأحمال في الشبكة اللاسلكية. وهو مستوحى من قوانين نظرية اللعب (Game Theory). النظام المقترح (DLBP) يعمل بشكل فعال على توظيف كل مستشعر جديد في الشبكة ليوزع عليه جزءاً من الحمل بشكل عادل بحيث لا ينحصر الضغط على جزء بسيط من المستشعرات.

بهدف تقييم الآلية المقترحة تم اختيار نظام مشابه ليتم المقارنة به وهو خوارزمية (Xue) حيث يعمل هذا النظام على توزيع الأحمال على الشبكة وذلك من خلال إجراء عملية حسابية في كل مستشعر قبل البدء بإرسال أية رسالة وذلك لمعرفة أفضل وجهة لتلك الرسالة. يقوم هذا النظام بحساب أوزان الأرسال للنقاط التالية ومن ثم حساب أفضل وجهة للإرسال ثم يرسل من خلال تلك الوجهة أو المستشعر ذا الاحتمالية الأعلى.

لقد تم إجراء عدد من التجارب لمقارنة النظام المقترح (DLBP) مع خوارزمية (Xue) وفقاً لعدد من المقاييس. تم دراسة توسع الشبكة ووجد أن نظام (DLBP) يزيد من قدرة الشبكة على التوسع عند زيادة عدد المستشعرات بنسبة تصل إلى 20% مقارنة مع خوارزمية (Xue). أما نسبة وصول البيانات المرسلة إلى هدفها بنجاح فوصلت إلى 97% أي أنها افضل من خوارزمية (Xue) بنسبة 16%. إضافة إلى ذلك تم تقليل الضغط الناتج عن إرسال بيانات لمعرفة الوجهة القادمة بنسبة تصل إلى 72%. كذلك تم خفض عدد العمليات الحسابية المعقدة بنسبة 99.9993%. كمحصلة لكل ما سبق فقد ازداد عمل الشبكة بنسبة تصل إلى 20% وهو الهدف المطلوب بداية من تطوير النظام.