QUERY BASED ENERGY EFFICIENT CLUSTERING METHODS FOR WIRELESS SENSOR NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

ONUR KOŞAR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

Approval of the Graduate School of Informatics

Prof. Dr. Nazife Baykal

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science/Doctor of Philosophy.

Prof. Dr. Yasemin Yardımcı Çetin

Head of Department

This is to certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Altan Koçyiğit

Supervisor

Examining Committee Members

Assist. Prof. Dr. Erhan Eren (METU, II) ------------------------------

Assist. Prof. Dr. Altan Koçyiğit (METU, II) ------------------------------

Assist. Prof. Dr. Banu Günel (METU, II) ------------------------------

Assist. Prof. Dr. Sevgi Özkan (METU, II) ------------------------------

Dr. Emrah Tomur (BDDK) ------------------------------

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Onur Koşar

Signature :

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude and appreciation to Assist. Prof. Dr. Altan Koçyiğit for his valuable support, professional guidance and suggestions throughout my thesis studies. I also thank him for his patience and encouragement during my studies.

I would like to thank my jury members Assist. Prof. Dr. Erhan Eren, Assist. Prof. Dr. Banu Günel, Assist. Prof. Dr. Sevgi Özkan and Dr. Emrah Tomur for reviewing and evaluating my thesis.

I am deeply grateful to my mother Makbule Koşar, my father Ahmet Koşar and my brother Erdem Koşar for their endless support and their love.

I would like to thank to my friends Yusuf Şafak Bayram, Hüseyin Bacanak, Mehmet Tosun for their valuable help and friendship.

Thanks are not enough to my colluages Evrim ÖZKÖK, Yaver CANSIZ, Sinan Can Açan, Ufuk Dumlu, Tayfun Asker and Uğur Adıyaman for their support in my studies and their valuable friendship.

I give my special thanks to Berrin Dıngaz who has been my biggest support through my thesis studies and my life. The words are not enough to appreciate her motivation, endless support and her lovely friendship.

ABSTRACT

QUERY BASED ENERGY EFFICIENT CLUSTERING

METHODS FOR WIRELESS SENSOR NETWORKS

Koşar, Onur
M. Sc. Department of Information System Technologies
Supervisor : Assist. Prof. Dr. Altan KOÇYİĞİT

May, 2011, 72 pages

In Wireless Sensor Networks, designing a low overhead routing protocol is crucial for prolonging network lifetime. Wireless sensor nodes depend on limited batteries and if they run out of battery, they cannot contribute to the sensing. There are lots of studies aimed at prolonging network lifetime. One of the methods to extend life time of the wireless sensor networks is clustering. In clustering approaches main aim is to prevent unnecessary messaging and decrease number of messages exchanged by aggregating messages. Clustering also contributes to prolong network life time by ruling the child node communications and therefore it decreases message loss caused by transmission collisions. Cluster heads in clusters schedule nodes for sending and receiving messages. In this thesis, a clustering approach based on queries disseminated by sinks is proposed. Two methods to prolong lifetime of sensor network by forming appropriate clusters and selecting suitable cluster heads is developed. Performance of the proposed methods is also evaluated with computer simulations.

Keywords: Wireless Sensor Networks, clustering, routing, network lifetime, wireless communication, Wireless sensor node

ÖZ

KABLOSUZ ALGILAYICI AĞLARI İÇİN SORGUYA DAYALI ENERJİ ETKİN KÜMELEME YÖNTEMLERİ

Koşar, Onur
Yüksek Lisans, Bilişim Sistemleri Bölümü
Tez Yöneticisi : Yrd. Doç.Dr. Altan Koçyiğit

Mayıs, 2011, 72 Sayfa

Kablosuz algılayıcı ağlarda en az enerji harcanarak haberleşmenin sağlanması, ağların yaşam sürelerinin arttırılmasında oldukça önemlidir. Ağ içerisinde yer alan algılayıcı düğümler pil kaynaklarının tükenmesi ile birlikte alan üzerinde algılama ve iletişim faaliyetleri dışında kalmaktadır. Kablosuz algılayıcı ağların yaşam sürelerinin arttırılması yönünde birçok araştırma yapılmakta ve yöntem önerilmektedir. Bu yöntemlerden birisi ise kümeleme yöntemidir. Kümeleme yöntemindeki ana amaçlar gereksiz mesajlaşmanın engellenmesi ve mesajların sadeleştirilerek mesaj sayısının düşürülmesidir. Kümeleşme ağ yaşam süresinin artmasına, kümeye dahil olan birimlerin haberleşme zamanlarını ayarlayarak, kablosuz ağlarda sıklıkla meydana gelen mesaj çarpışmaları sebebi ile gerçekleşen mesaj kayıplarının önüne geçerek yardımcı olur. Küme başları küme içinde yer alan birimlere mesaj gönderme ve alma zamanlarını belirterek onları yönetmektedir. Bu çalışmada sorguya dayalı bir kümeleme yöntemi önerildi. Uygun kümleler oluşturarak ve uygun küme başı seçerek algılayıcı ağın yaşam süresini artırmak için iki yöntem önerildi. Önerilen yöntemlerin başarımı bilgisayar benzetimleri ile değerlendirildi.

Anahtar Kelimeler: Kablosuz Algılayıcı Ağlar, kümeleme, yönlendirme, Ağ yaşam ömrü, Kablosuz iletişim, Kablosuz algılayıcı birimi

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

CHAPTER 1

INTRODUCTION

People need to monitor changes in environmental conditions for variety of purposes. Extracting data from changing environment and interpreting that data to gain reasonable information enable people to make meaningful decisions. Today, automation of data collection is facilitated by the improvements in computation and wireless communication technologies. In order to monitor environment or systems, low cost computation and communication devices have been developed. Those devices have sensing ability with built in sensors, basic computational facilities and wireless communication capabilities. With the advances in wireless networking technology, a wireless sensor network can be deployed without a fixed infrastructure. Nodes in the network connect to each other in ad-hoc fashion and they communicate according to wireless communication standards such as; 802.11, Zigbee, 802.15.4, ISA 100.11.a etc. In such networks which are called wireless sensor networks, since they depend on wireless communication, it is possible to tolerate individual node failures, and new nodes can join/leave the network during the normal operation of the network without requiring a total reconfiguration.

Wireless sensor networks are widely used and preferred for environmental monitoring, military applications, health care, industrial monitoring, etc. Wireless sensor networks consist of different kind of interoperable nodes distributed in an area and those nodes employ wireless communication. By using flexible communication and routing schemes it may also be possible to add/remove nodes into/from the network while it is operating. For example, in order to recover from node failures affecting monitoring quality/coverage, new nodes can be deployed on to the sensing region and after a negotiation phase, new nodes can start to contribute sensing process. This capability adds flexibility to enlarge sensing area and also it contributes to the extending network life time.

In wireless sensor networks, each sensor node mainly consists of a radio transceiver, a micro controller and an energy source generally a battery unit. Nodes usually have limited battery energy and the lifetime of the network is closely associated with the energy consumption rates of the nodes. Nodes consume most of their energies while sending and receiving messages. There are lots of studies focusing on extending the lifetime of wireless sensor networks. The primary method to extend lifetime is to decrease the number of messages produced/transmitted by each node. In conventional sensor networks, all nodes send their messages to the sink node whenever they detect an event. If they are not within the range of the sink, they depend on other nodes to relay their messages towards the sink node. This causes transmission of too many messages and therefore lifetime of the network decreases. One of the methods to decrease the number of messages exchanged is clustering. In clustering, one of the nodes among a set of sensor nodes is selected as the cluster head. In clustering case, nodes in the cluster do not send messages directly to the sink. They send their messages to the cluster head and cluster heads collect and aggregate received messages into one possibly larger message and send it towards to the sink node. Clustering provides tremendous decrease in total number of messages flowing in the network. There are many approaches proposed for proper clustering in wireless sensor networks. Some of those approaches are:

ι. Dedicated cluster heads and cluster members: In this approach nodes are designated as cluster heads or cluster members before the deployment phase. In general, cluster head nodes are equipped with larger batteries compared to cluster members. Clusters are formed during the deployment and usually they are not reconfigured.

ιι. Dedicated cluster heads: In this approach some nodes are designated as cluster heads before the deployment. Cluster heads form clusters during normal network operation by selecting cluster members according to application needs.

ιιι. Randomly selected cluster heads: In this approach, each node has a chance to be a cluster head. Nodes are elected as cluster heads according to some sort of algorithms and each node has a chance to be cluster head according to conditions defined in algorithms. LEACH and C-LEACH are examples of algorithms that employ randomly selected cluster heads.

ιϖ. Selecting cluster heads using semantic definitions: Cluster heads are selected as semantic definitions carried in query messages. Nodes process messages and applies query in the message to their sensed data. That is, nodes decide becoming cluster head or not according to the definition in the received query message.

1.1 Problem Definition

There are lots of clustering algorithms proposed in the literature. Beside application specific clustering schemes, the main aim of clustering is to reduce energy consumption of nodes in the network by means of removing unnecessary messaging between the nodes and the sink node. In flat WSN topologies, each node that detects an event immediately sends a message to the sink via one hop or multi hop communication. However, this approach produces a huge number of messages flowing in the network and heavy usage of communication medium results in collisions that further cause waste of network capacity. Moreover, collisions lead to retransmissions and therefore, node's energy is wasted.

Clustering approach offers data aggregation and decreases total messaging needs compared to the flat topologies. Data aggregation is done at cluster head nodes and by means of data aggregation the number of messages transmitted and relayed to sink is considerably decreased. There are different cluster formation techniques, and selection of appropriate clustering technique depends on the purpose of clustering and the requirements of the application. Cluster head nodes can be located and elected before implementation phase and they can be stationary during the network life time, they can be selected randomly, they can be selected in a predefined order or they can be selected as using semantic definitions. Those clustering mechanisms can be classified as; static which provides local topology control, dynamic which results in forming clusters according to changing network parameters. Moreover, clustering mechanisms can also be classified according to communication mode such as single hop and multi hop and node type which are homogeneous and heterogeneous.

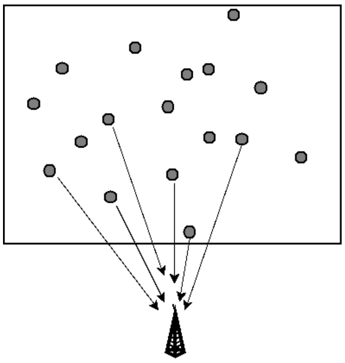

Figure 1.1: Single hop mode

In one hop mode, whenever a node in the network detects an event, it sends a message to the sink node directly (Figure 1.1). As transmission energy consumption increases with the distance between nodes, one hop communication is a very expensive operation.

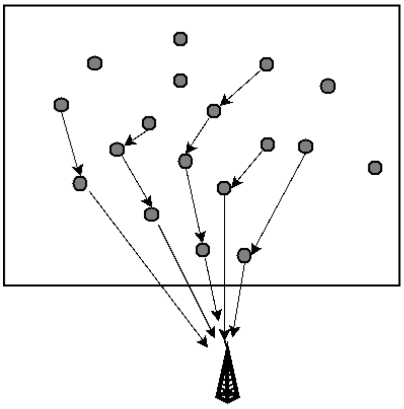

Figure 1.2: Multi hop mode

In multi hop communication mode, nodes send messages to the neighboring nodes that are in the range of their transmitter and intermediate nodes relay messages to their neighbors until message is delivered to the sink node (Figure 1.2). Since nodes transmit messages in a limited range, this sort of communication results in nodes to efficiently use their energy compared to the one hop communication mode. However, as opposed to one hop communication mode, in multi hop communication mode the same message is transmitted repeatedly by the neighboring nodes towards the sink node, therefore we have to take into account the additional energy spent by intermediate node while relaying messages. Since the energy consumption rate depends on the transmission power and required transmission power grows exponentially with the distance between the nodes, in some cases, this approach is expected to be more energy efficient than the single hop communication mode.

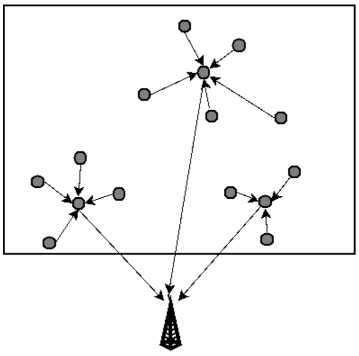

Figure 1.3: Clustering with one hop communication

In clustering approaches, cluster head nodes transfer messages of child nodes to the sink node. Cluster head node collects messages that are sent from the child nodes, makes an aggregation operation, prepares an aggregate message and sends it to the sink node. According to the design of the network, cluster head node can send message to the sink node with one hop communication as shown in Figure 1.3.

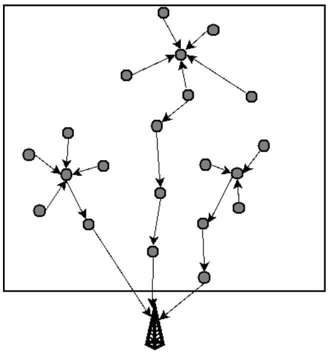

Figure 1.4: Clustering with multi hop communication

Cluster head nodes can also send their messages to the sink node through multi hop communication (Figure 1.4). In that case, cluster head nodes collect the messages of the child nodes in the cluster and apply an aggregation operator on those messages and send the resultant message to the sink via non-cluster head nodes in its sending range.

Semantic clustering is another method to form clusters according to the information carried in the query messages. Cluster formation is triggered by the query message created by the sink node and cluster formation starts when the first node that satisfies the condition in the query message

[1]. In this approach the node that assign itself as cluster head starts cluster formation with disseminating cluster formation messages in its range. However, such a cluster head selection method does not consider remaining energies of the nodes and the location of cluster heads during cluster formation. In this selection, the nodes that start cluster formation may not have enough energy to support cluster communication and that situation potentially causes improper cluster formations.

1.2 Objectives of the thesis

There are diverse semantic clustering approaches designed for WSNs. The approach stated in [1], offers a model in which the nodes change their behaviors according to the message they received.

Four types of messages defined in the model are: [1]

i. Advertisement message is used for interest propagation, initialized by sink node,

ii. Join message is used by the child nodes to join the cluster and produced by cluster head nodes,

iii. Join advertisement message is used for announcing clustering information for second and third level child nodes,

iv. Join advertisement request message is used by second, third or upper level child nodes to join parent nodes and it is produced by the higher level parent nodes in clusters.

In [1], the first node which receives an advertisement message and meets the condition stated in the query in the message announce itself as cluster head and starts cluster formation. Cluster formation process continues until nodes cannot have data to meet the query stated in message sent from cluster head or parent nodes. Cluster depth is not controlled in the model, and cluster heads’ physical conditions are not considered during cluster formation phase.

In this thesis, a new cluster formation phase is designed for selecting cluster heads that take into account the energy levels and locations of the nodes in the cluster. This phase takes place after the cluster formation is completed in [1]. New message types for cluster head selection and messaging between the nodes are defined. Cluster head re-selection models that we propose consider energy efficiency in intra cluster communication. For this purpose beside the proposed model I described in [1], we propose two additional algorithms to select the cluster head, selecting the node which is close to the gravity center of the nodes in the cluster and selecting the node which has the highest battery level in the cluster. We also consider minimizing the number of messages produced in intra cluster communication to prolong life time.

In order to evaluate and compare the performances of proposed methods a computer simulation software is developed. Various scenarios and topologies are simulated and results are discussed.

1.3 Thesis Organization

The remainder of this thesis is organized as follows: Chapter 2 includes literature review on the related work and provides background information on wireless sensor networks. In addition, routing protocols used in WSNs, medium access control mechanisms, clustering approaches and semantic clustering methodologies are discussed.

Chapter 3 introduces the network model considered in this thesis as well as the proposed re-clustering approaches.

Chapter 4 includes detailed information about the simulation work. This chapter introduces the simulation software developed. Then, proposed models were tested in simulation environment and network performances are evaluated. Results obtained from simulations on different network topologies and scenarios are presented and discussed. Results of simulations are also provided in Chapter 4.

Chapter 5 outlines conclusions about the work done and suggests possible directions for future work.

CHAPTER 2

LITERATURE REWIEV

Advances in communication techniques and electronics led to improvements in WSN designs and implementations. Recently, Lots of researches have been done on WSNs. The key concept of increasing life time of the sensor networks have been taken into consideration in many studies. In this chapter, review of the related works in literature is mentioned. Chapter consists of ; WSN definitions and usage (Section 2.1), routing protocols in WSN (Section 2.2), Medium Access Control mechanisms in WSN (Section 2.3), clustering approaches in WSN (Section 2.4), semantic clustering and query processing in WSN. (2.5).

2.1 WSNs and Their Usage

WSNs consist of nodes that can communicate with each other over wireless communication channels and intended to monitor some phenomena in a physical area. Nodes in the network mainly consist of a sensing unit, a processing unit, a transceiver and a power unit [2] [27]. The nodes are equipped with a processing unit, therefore nodes can do some basic computations on sensed data and they send the sensed information to a special node called sink. Sensor networks can be deployed in inaccessible areas with unpredicted location of nodes. This necessitates the sensor nodes to organize themselves to form a WSN and perform routing across the network in an organized way. Nodes in the network depend on their batteries. Therefore, energy efficient operation is very crucial to increase the life time of the network.

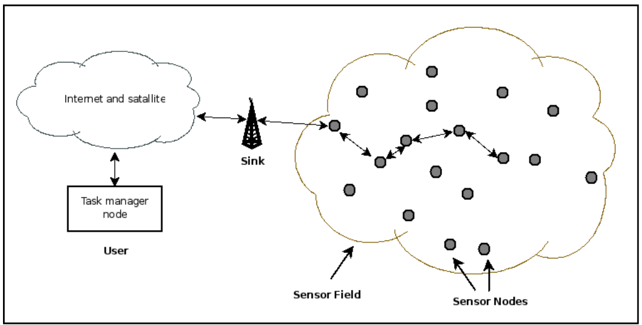

Figure 2.1.1: Sensor nodes scattered in a sensor field (reproduced from [2])

Sensor networks observe phenomena in their sensor fields and user is noticed instantly. A node in the network prepares a message towards to the sink node and the sink node directs message to the user via a communication medium such as; the Internet, satellite etc. Fig. 2.1.1.

In most of the WSN deployments there are not any physical dedicated connections between nodes. Nodes in the networks transmit their data in an ad-hoc manner. In general, sensor nodes use wireless medium to communicate and this provide enlarging the network by means of adding new sensor nodes to the network. Node failures can also be tolerated with new node additions. This flexibility results in covering the sensing area more efficiently and provides robust observation conditions. Sensor networks are used in a variety of fields. Beside military applications and area monitoring, sensor networks are also be used in health industry. There are commercial applications that monitor patients’ health conditions with some application specific sensors. One of the other usages of sensor networks is for developing smart home applications. One can use sensor networks in homes to control heat level, electricity consumption of the devices, fire detection etc.

2.2 Routing Protocols in WSN

Nodes in WSN are connected to each other by means of wireless channels. Information is disseminated through the network in an ad hoc manner since there is not any dedicated direct link between the nodes. Several different routing protocols are proposed for WSNs. Those protocols can be classified as data-centric, hierarchical, location based. Some protocols also consider network flow and quality of service [3].

Akkaya and Younis (2005), state that data sending methods can be categorized into continuous, event-driven and hybrid. According to the requirements and the architecture of the WSN one of these sending methods can be used. In general, routing protocols can be classified as; data-centric, hierarchical and location based. In addition to those routing approaches, in some routing protocols network flow and quality of service are also taken into consideration. One of the important considerations for routing information through the network is eliminating transmission of redundant data. There are some proposals to reduce redundant data transmission. For example, SPIN (Sensor Protocols for Information via Negotiation) is a data-centric routing protocol that eliminates redundant data transmission and offer energy efficient routing in the network.

Kulik, Rabiner and Balakrishnan (2008) state that comparing SPIN with the conventional routing approaches reveals that for the same amount of data, SPIN performs, %60 better sending ratio. In this protocol, meta-data packages are delivered through the network in communication.

This meta-data describes the intended information. Routing paths of the data is determined with considering the meta-data packages. In WSNs there are some data delivery methods impose diversity in applications. Data produced by sensor nodes can be delivered to the sink node continuously, event-driven, query-driven and hybrid [3]. In the model of continuous delivery, nodes send their data periodically without any stimulus. In event-driven and query-driven data delivery models nodes are triggered with events or they response to queries that they receive [28].

Routing protocols in WSNs can be classified generally as data-centric, hierarchical and location-based [3]. In data-centric routing applications, it is intended to find the specific nodes that meet the conditions without propagating all of the nodes in the network. It differs from traditional address based routing approaches in attribute-based naming are used. In data-centric routing, nodes negotiate to each other in order to eliminate redundant data occurrences. Some example of data-centric routings can be listed as; flooding, gossiping, SPIN, Directed Diffusion, Rumor routing which a variation of Directed Diffusion, Gradient-based is routing, ACQUIRE. Hierarchical routing protocols provide efficient energy consumption with multi-hop communication and data aggregation [29]. LEACH [30], is one of the most popular hierarchical routing protocol used in WSNs. This hierarchical routing scheme provides clustering of nodes and uses local cluster heads as work as routers to the sink. LEACH approach offers random change of cluster heads in order to balance energy dissipation among nodes.

In our proposed model we use query-driven data delivery method in the network. A query is disseminated through the network and clusters are formed according to the condition stated in the query. Details of Semantic clustering and studies on this approach are described in 2.5.

2.3 Medium Access Control Mechanisms in WSN

WSNs communicate to each other on wireless communication channels. IEEE 802.11, Zigbee, IEEE 802.15.4, ISA 100.11.a protocols are commonly used physical and MAC layer protocols. Since communication medium is shared with the nodes in transmission ranges, some medium access mechanisms should be used to coordinate transmissions of the sensor nodes. CSMA-CA and wireless TDMA based protocols are preferred in most WSN applications. In some cases combination of the both approaches are employed for application specific purposes and for mainly energy efficiency in communication.

Snow and Fend (2009) proposes an energy efficient medium access protocol for wireless sensor networks and they commented that a combination of MAC protocols can be developed according to the needs of the intended application. It is also stated and evaluated that pre-scheduling the medium access with hybrid MAC protocol usage has better energy efficiency as compared to other non-hybrid MAC approaches [8]. In our model a hybrid medium access mechanism was used. In the deployment, TDMA medium access mechanism is used in intra cluster communication and CSMA/CA mechanism was used for inter cluster communications. Selection of medium access mechanisms depend on network design and application. It should be considered that TDMA scheduling performs better because assigned time slots and scheduling transmission of nodes prevent collisions; however, all of these operations require time synchronization and extra messaging overheads. If the design of the network is not likely cause collision of network packages, CSMA/CA scheme could be deployed.

In respect to network conditions, hybrid CSMA/TDMA medium access mechanisms outperforms in energy consumption comparing with non-hybrid mechanisms.

2.4 Clustering approaches in WSN

Clustering is the commonly used method in WSN for better information negotiation performance and to reduce communication overhead. The main aim in clustering is providing energy efficiency in communication. A lot of studies have been carried out for energy efficient routing protocols and some of them are based on clustering. There are many protocols designed to reduce the network traffic toward the sink node. In clustering nodes are registered to a cluster head at the beginning and cluster head coordinates the communication within the cluster. Therefore, clustering also causes some overheads as additional phases of cluster formation and cluster maintenance is necessary. However, it has been shown in [10] that clustering based approaches performs better in energy consumption comparing with flat network topologies in WSNs.

Table 2-5-1, Clustering routing protocols and feature comparisons (reproduced from [10])

	Periodic, Event-Query-based	Single Hop	Qos	Sink engaged in cluster formation	Data aggregation	Only CHs are relay node
LEACH	*	*			*	*
TEEN				*	*	*
APTEEN	*		*		*	
CPEQ	*				*	
ICE	*		*		*	

Commonly used Clustering mechanisms and their specifications are compared in Table 2-5-1.

2.5 Semantic Clustering and Query processing in WSN

Semantic clustering is a clustering approach that considers the semantic definitions which is carried on messages produced by nodes. In general a query mechanism is used in semantic clustering algorithms. Nodes form clusters according to the conditions stated in queries. Cluster life times are application specific and nodes destroys cluster after a predefined time period. Choosing optimal clustering algorithm requires considerations of state of following parameters; battery level, processor load, transmission power and node degree [26]. Querying mechanisms in semantic clustering is also an important issue in deployment. In query processing, a query is disseminated through the network, each node receives the query implements a loop of operations [24]. In this loop, nodes process the query, computes and evaluate the results and send those results to the sink node. Sink node collects the messages of the queries and aggregate those data to conduct meaningful information. Decreasing number of messages resulted from query response messages by means of message processing and aggregation in network provides better results comparing with the traditional one-hop communications. He and Tully, 2008, proposes in distributed database approach for query processing in WSNs, and this approach provides better in network data aggregation and reduce in messaging which are crucial for network life time. There are lots of studies that uses query processing to form clusters which are consisted of related nodes that meets the query.

Hu and et al, suggest a querying mechanism that forms clusters accordingly and they provides results that performs better energy dissipation ratios comparing with LEACH model. In our proposed models we considered semantic clustering method used in [1]. We proposed additional signaling and a re-clustering phase in order to select more appropriate cluster head in the cluster formed by query diffusion.

CHAPTER 3

NETWORK MODEL AND PROPOSED ALGORITHMS

3.1 Network Model

The network considered in this work consists of N nodes. Nodes are deployed on to a sensing region randomly. The single destination node (sink node) is assumed to be placed at the within the sensing region. Nodes are stationary, and they operate until they run out of battery. All sensor nodes except the sink node have the same amount of energy when they are deployed. Sink node is assumed to have an unlimited power supply. In addition, all sensor nodes are identical and they all have built in sensors required by the application and they all have GPS module installed for location discovery. Nodes are capable of remembering log of processed messages, and other cluster members together with their locations. In this network, any two nodes are said to be neighbors if they are within the communication range of each other and they have enough energy to communicate.

Nodes in the network have the same transmission capacity. Nodes receive messages if they are in sending range of the sender and if they have enough battery power to receive the message. Nodes are removed from the network if they run out of battery. Once clusters are formed, new nodes cannot join the existing clusters in the network. That is, cluster maintenance is not performed for new node additions. There are two medium access mechanisms employed in the network, one of them is TDMA and the other CSMA/CA. Intra cluster communications are performed according to the TDMA and cluster head to sink communication is performed by CSMA/CA. Intra cluster communication is divided into rounds in which TDMA schedule is employed. Therefore, each node is assigned to a time slot for communication to prevent collisions.

Query dissemination is performed by flooding the message to the medium. First of all, sink starts the flooding by sending query message to the nodes in its range. Messages are delivered through the network in ad-hoc fashion and message types change according to decisions given by the nodes after processing messages. There are four main message types used for clustering in the network. In the first phase, sink node prepares an advertisement message to extract information through the network. Advertisement message is used for finding nodes that satisfies the information which is intended to be extracted.

In the cluster formation phase, each node has a chance to be a cluster head, therefore nodes have also ability to do calculations as each cluster head can do. Nodes are assumed to have limited sensing and transmission ranges. Messages are disseminated through the network by means of neighbor communications. In normal operating conditions, nodes usually have a circular range and nodes can communicate with the neighboring nodes if they are in sending range and they have enough battery. Nodes are equipped with limited power sources at the beginning of the deployment. Initially, all nodes except sink node have a single battery unit by default. A node's battery level decreases in time constantly, because sensing and listening continuously consume energy.

The transmission cost is proportional to the distance between the two communicating nodes and it is calculated as in section 3.2.

There are three types of communications in the WSN. In the query dissemination phase, sink node floods the query message and the message is disseminated in the network. After cluster formation, child nodes send their messages to their parent nodes by node to node communications. Moreover, cluster heads sends their messages to the sink directly with one hop communication. Cluster head nodes use a different channel for cluster head to sink communication. This channel is used by cluster head nodes and normal type nodes cannot use the channel. In order to reach the sink node with one hop communication, cluster head nodes adjust their transceivers to an appropriate range. All nodes are capable of using different channels for intra-cluster communication and cluster head to sink communication to avoid interference. While forming clusters, a proper channel and a TDMA schedule to use within the cluster are assigned by the cluster head. Cluster head to sink channel is fixed and that channel is used while forming clusters (during query dissemination) and while cluster heads transmit their aggregated data to sink. Although cluster heads use two different channels during data collection and transmission to sink, since inter-cluster and intra-cluster communications are performed in non-overlapping time periods, a single transmitter is enough for each node.

3.2 Node Properties

Nodes in the network have capability to sense in the region defined by their sensing range. Nodes also have sending range defined by the default. Sending range (SR) of the nodes depends on the transmission power set at the beginning. Nodes are assumed to send messages to the medium in a circular region; the area of this region is related to the Sending Range (SR) of a node and defined as $\pi \times \mathrm{Rts}^2$ where Rts is the transmission range of a node. In that range neighboring nodes which listen to the medium can hear the messages in the medium and receive message to be processed. Each node has a single battery unit with limited initial energy and this battery level decreases constantly because of sensing and processing costs. In addition to the constant decrease, each node's battery level is decreased while sending and receiving messages. Nodes which run out of battery are said to be dead nodes and they cannot contribute to sensing and routing. Processed message identities and related queries in messages are stored in nodes for new message comparisons and for extracting past query search results. Nodes store child node and cluster information during the cluster lifetime.

Nodes have two different transmission modes: intra cluster communication mode and cluster head to sink communication mode. Nodes use different communication channels and adjust their transmission power according to the current transmission mode. In intra cluster communication mode, nodes use lower transmission power compared to transmission power used in cluster head to sink communication mode. In cluster head to sink mode nodes' transmission powers are adjusted to reach the sink in one hop. A predefined control channel is used to transfer messages from cluster heads to sink. That control channel is also used during query dissemination and cluster formation (or interest propagation) phase but in these phases intra cluster communication mode is used. For intra cluster communication mode, the proper channel chosen by the cluster had to avoid interference from neighboring clusters is used by the nodes.

3.4 Proposed Model I

3.4.1 Interest Propagation

In Proposed Model I the method that is proposed in [1] is investigated. Interest message is created by the sink node at the beginning of the deployment. Sink node prepares an Advertisement type message which contains a query to be processed. Advertisement message consists of message ID, Sender Node ID, query, round time, round count and path of the message. This message is flooded through the network by the sink node. In interest propagation phase, nodes access the medium using the non-persistent Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol over the control channel.

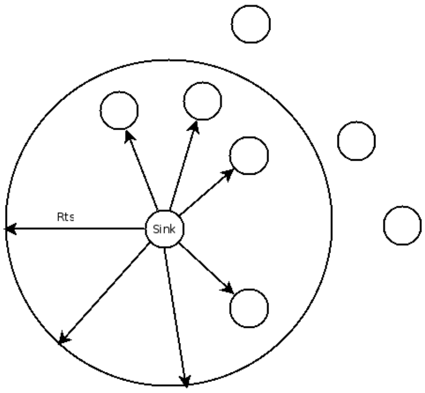

Figure 3.4.1.1: Node's Sending Range

Message is first sent to the nodes in the transmission range of sink. Neighboring nodes in sink’s range can hear the message if they have not run out of battery. Query in advertisement message is extracted by the receiving nodes and the query is applied for their data. Nodes start cluster formation phase if they receive an advertisement message and the result of the query is true. Otherwise, the same advertisement message with the same message ID is relayed to the other nodes in node's range. When nodes relay the message to the other nodes, each node adds itself to the path variable in the message.

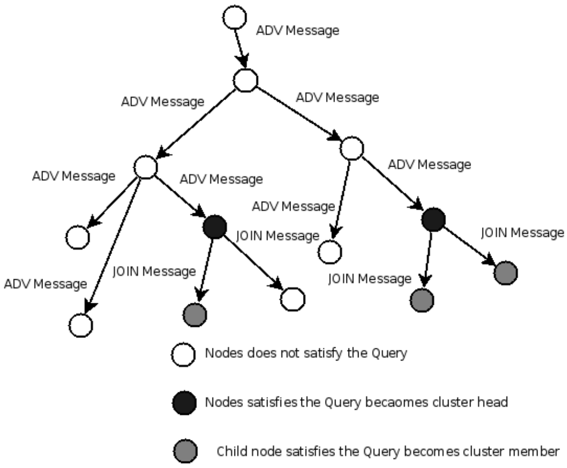
ADV Message
ADV Message
ADV Message
ADV Message
ADV Message
ADV Message
ADV Message
JOIN Message
JOIN Message
ADV Message
JOIN Message
JOIN Message
○ Nodes does not satisfy the Query
● Nodes satisfies the Query becaomes cluster head
● Child node satisfies the Query becomes cluster member

Figure 3.4.1.2: Interest Propagation

In interest propagation, nodes which do not satisfy the query just add themselves to the path variable in the message and relay the message to the neighboring nodes in their sending ranges. Message ID of the first message which is produced by the sink node is not changed until message type is changed by nodes. Nodes remember the message ID's of the messages they processed before in order not to re-process the same message again. Nodes drop messages if they have already processed or received the message. Nodes which satisfy the query changes the message type advertisement to join message and start cluster formation phase.

3.4.2 Cluster Formation

Nodes become cluster head if they receive an advertisement message and if they are able to respond to the query in the message. Cluster head node creates a new join message to start cluster formation. Join message consists of Cluster head ID, query, path and type of the message. Join message is first sent to the nodes in cluster head’s range. Nodes which satisfy the query may decide to join the cluster if they haven’t received any other join message. Negotiation between nodes for cluster formation continues until a predefined level child node joins the cluster.

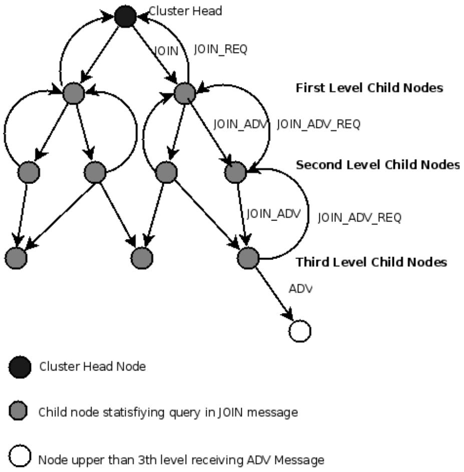

Figure 3.4.2.1: Cluster Formation

First level child nodes are described as the nodes in the sending range of cluster head. Second level child nodes are nodes in the sending range of first level child nodes and third level child nodes are nodes in the sending range of second level child nodes. In Figure 3-3, a sample cluster formation messaging is illustrated for a maximum level of 3.

Table 3.4.2.1 Processed messages table

Message ID	Message Type	Time

First level child nodes which receive the message and check whether they have received the message before or not by searching message ID in their processed messages table (Table 3-1).There are three kinds of messages used in cluster formation. Join is used for starting of cluster formation phase. First level nodes of cluster head hear the join message and they apply the query in the message to their data. If they are able to respond to query, first level nodes prepare a join request message and send this message back to the cluster head. Cluster head receives those messages and updates its joined nodes table accordingly.

Table 3-2, Joined Nodes table of cluster head

Message ID	Message Type	Sender ID	Time

Cluster head node first checks whether received join request message is the reply of the join message which is sent by the cluster head. This control is done by controlling the message ID and receiver ID in the join request message. If the receiver ID of the join request message is not equal to the cluster head’s ID, cluster head drops the message. Otherwise, accepts the message and updates the joined node's table (Table 3-2). Cluster head node does not process the messages of join request and advertisement during the cluster’s life time. Joined child nodes of the cluster head creates a join advertisement type message from the received join message and transmits that message in their ranges. Child nodes join to the cluster if the message type is join advertisement and the result of application of query to their data is true. Child nodes that want to join to the cluster creates an join advertisement request message with Node ID, Parent ID, Message ID and send this message to the parent Node. Parent node updates its joined node table according to the received join advertisement request messages. The same procedure is repeated for the next level child nodes are executed. Every parent node saves child node information in their joined nodes table until cluster is destroyed.

Cluster depth is controlled by a depth count field in join and joins advertisement messages. Depth count is initialized to '0' when the first join message created by the cluster head. The value is increased when child nodes create join advertisement messages. Child nodes that receive a join advertisement message check the depth count. If the depth count is 3, child node creates a join advertisement request message and sends this message to the parent node in order to join the cluster. In addition, third level child node creates an advertisement message with the same message ID, and sends this message to the nodes in its range.

Cluster head node knows only the first level child node information. Each parent node stores the child node information of them. Parent nodes wait for a predefined time limit (parent_wait_time_limit) for children nodes to join the cluster. After the predefined time, parent node composes an information message which contains children node count of the parent node and sends this message to the cluster head. Cluster head node waits up to predefined time limit (clusterhead_wait_time_limit) and then calculates total node count of the cluster by adding node count information gathered from parent node's information messages.

3.4.3 Inter Cluster Communication

There are two kinds of values that determine cluster lifetime. Those values are round count and round time. The lifetime of a cluster is defined as round_count x round_time. The values are specified in the first advertisement message which is created by the sink node and that information is propagated into the network. Cluster head nodes are triggered by the completion of cluster head waiting time limit. Cluster heads then calculates the total node count by extracting the count information gathered from nodes in the cluster.

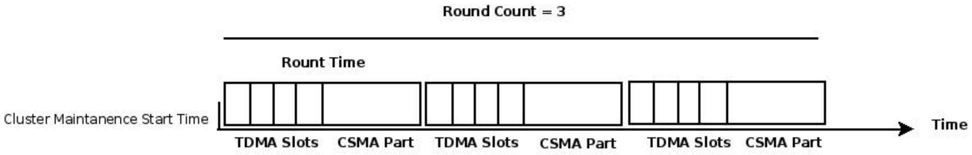

Figure 3-3-3-1, Communication phases in cluster maintenance

Two types of communications are performed in cluster maintenance phase. One of them is inter cluster communication and the other one is intra cluster communication. Intra cluster communication is performed by means of Time Division Multiple Access (TDMA) mechanism. In this period cluster head assigns time slots to the nodes in the cluster. Inter and intra cluster communications are performed for time duration of round_time/2, each.

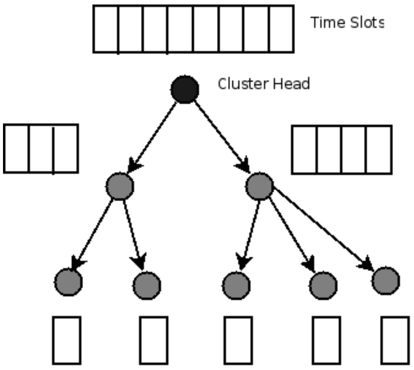

Figure 3-3-3-2, Time Slot assignment in cluster maintenance

Cluster head nodes start assigning time slots to the child nodes after the cluster formation phase is completed. Cluster head node divide intra cluster communication time to cluster node count and assign time slots to the parents proportional to the child counts (Figure 3-3-3-2).

Cluster head node prepares a slot assignment message and sends that message to the parent nodes. Slot assignment message consist of the type, message ID, parent ID, and series of node_ID-assigned_slot two-tuples. Parent node extract the information in the slot assignment message, saves the slot of it and prepare new slot assignment messages for the child nodes. Slot assignment messages prepared by the parent nodes consist of type, parent ID, node_ID, assigned_slot. Child nodes receives the message and checks for the node ID, if the message was sent to it, it assigns that slot for its messaging.

The last slot of the intra cluster communication is assigned to the cluster head. Cluster head's receivers are always kept on and it listens to the medium constantly. After the end of intra cluster communication, cluster head aggregates the collected messages into one message. This message consists of the sender ID and information. The cluster head to sink message consists of node locations and node ID two-tuples that corresponds to the nodes which has data to meet the query in the advertisement messages. At the end of the rounds cluster head nodes send aggregated message to the sink node directly with one hop communication over the control channel using CSMA/CD protocol.

After the cluster head life time ends, all nodes in the cluster clears their joined node tables and cluster heads start to work as a normal node.

3.5 Proposed Model II

Proposed model II differs in the proposed model I with re selection of cluster head after the cluster formation phase is completed. After cluster head is selected and cluster is formed, cluster head assigns a new cluster head by considering the locations of the nodes in the cluster.

The main aim in selecting new cluster head is to find node which is closer to the average X and average Y coordinates of the nodes in the cluster.

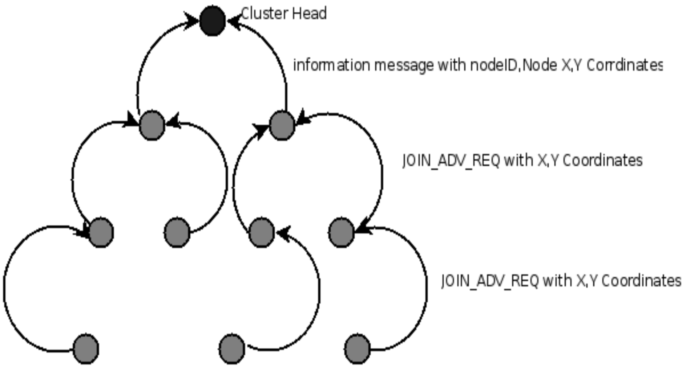
Cluster Head
information message with nodeID,Node X,Y Corrdinates
JOIN_ADV_REQ with X,Y Coordinates
JOIN_ADV_REQ with X,Y Coordinates

Figure 3-4-1, Messaging in proposed model 2

During the cluster formation phase; parent nodes gather child node's X, Y coordinates with the join advertisement request messages. Child nodes add their X, Y coordinates to the join advertisement request messages. Parent nodes wait for child node's responses up to parent_ wait_time_limit and then prepares an information message which contains sender ID, cluster head ID, joined nodes ID – Joined node's X,Y values couples with parent node X,Y value and node count information. Cluster head waits up to clusterhead_wait_time_limit and stores parent node information messages. After waiting for parent information messages, cluster head node calculates average X and average Y values of the cluster members. Then it makes a comparison with node's X, Y values with the average X, Y values and selects the node which has nearest to the average X, Y values. The node which has closest Euclidean distance to the average X, Y value is selected as the cluster head. After selecting the new cluster head, current cluster head prepares a cluster head advertisement message and sends this message to nodes in its range. This message consists of message type, new cluster head ID and list of node Ids in the cluster. First level child nodes which received the cluster head advertisement message propagate the message to their child nodes. Parent Nodes get the node list for the cluster and store this information for re clustering phase. After propagating new cluster head advertisement message each node clears their joined nodes table. Message is propagated until it reaches the new cluster head node. Nodes compare their Ids with the new cluster head's Id when processing the message and after processing they clean their joined nodes table. New cluster head node also clears its joined node table and prepares a new advertisement message and sends this message. The message type of this message is defined as re cluster advertisement message.

Cluster formation mechanism and messaging in this step are the same as proposed model 1 except that nodes checks the node list of the previous cluster in order to control only nodes that were participated in clustering are included in new cluster formation. After cluster formation is completed. Cluster head node assign TDMA schedules to node as described in proposed model 1.

3.6 Proposed Model III

This model considers nodes’ battery levels when re selecting cluster head. The only difference in this model is assigning new cluster head after cluster formation as the node which battery level is the highest. The first cluster head selection mechanism and the related messaging are the same with the proposed model 1.

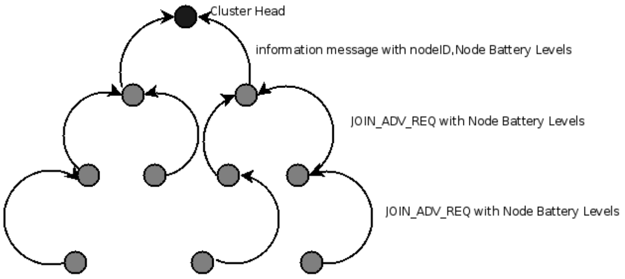
Cluster Head
information message with nodeID,Node Battery Levels
JOIN_ADV_REQ with Node Battery Levels
JOIN_ADV_REQ with Node Battery Levels

Figure 3-5-1, Messaging in proposed model 3

During the cluster formation phase; parent nodes gather child nodes’ battery levels with the join advertisement request messages. Child nodes add their battery levels to the join advertisement request messages. Parent nodes wait for child node's responses up to parent_ wait_time_limit and then prepare an information message which contains sender ID, cluster head ID, joined nodes ID – Joined node's battery levels couples with parent node battery level value and node count information. Cluster head waits up to clusterhead_wait_time_limit and stores parent node information messages. After waiting for parent information messages, cluster head node calculates the highest battery level and matches this battery level with the node Id. The node which has the highest battery level is selected as the new cluster head. After selecting the new cluster head, cluster head prepares a cluster head advertisement message and sends this message to nodes in its range. This message consists of message type, new cluster head ID and list of node Ids in the cluster. First level child nodes which receive the cluster head advertisement message pass the message to their child nodes. Parent Nodes gets the node list for the cluster and store this information for re clustering phase. After sending new cluster head advertisement message, nodes that process the message clears their joined nodes table. Message is relayed until it reaches the new cluster head node. Nodes compare their Ids with the new cluster head's Id when processing the message and after processing they clear their joined nodes table. New cluster head node also clears its joined node table and prepares a new join message and sends this message to the nodes in its range. This message is used to form new clusters by means of advertising the new cluster head and inviting neighboring nodes to the cluster if they meet conditions stated in message.

Cluster formation mechanism and messaging in re clustering phase are the same as it is done in proposed model I except that nodes check the node list of the previous cluster in order to control that only nodes that were participated in clustering are included in new cluster formation. After cluster formation is completed. Cluster head node assign TDMA schedules to node as described in proposed model I.

CHAPTER 4

SIMULATION STUDY

In this study, a computer simulation software has been implemented to evaluate the performance of the proposed models. The simulator was developed in Java programming language and Netbeans integrated development environment was used for development. In order to evaluate the performance, network life time, total battery consumption in time, number of living nodes when simulation is completed, total messages produced, clusters formed are measured for a set of sample scenarios. The simulator developed saves the simulation traces in a database. We used postgreSQL database management system for this purpose. Results of each simulation run are saved in their related tables and meaningful information extracted from these traces.

We consider three different energy consumption calculations in simulation. There is a constant decrease in battery levels of all nodes at each simulated unit time. This energy consumption is caused by the node's sensing processes of the field and listening process to medium for incoming messages. Nodes consume energy when they receive and process messages. This cost is calculated as E receive = ρ.L [31], for receiving L bits packets. ρ is the coefficient of power consumption and it is used as 50 nJ/b and E stands for energy consumption.

The energy consumed while sending L bits message between two nodes is:

E transmission = $\alpha+\beta \times d^{m} \times L$ [31]

Where α and β are constant values, d is Euclidean distance between nodes, L is the number of bits to be transmitted, and m is the path loss exponent in range 2<= m<=4. Two constants are taken as   α=50 nJ/b and β=10pJ/b/m2 (for m=2). In our model m is taken as 2.

4.1 Simulation Software

Simulation software consists of: Node, Message, Entry, GraphicsUtil and a main class.

4.1.1 Node Class

Node is the main element of a wireless sensor network. Node class is used to represent real wireless sensor nodes. Nodes generally consist of a transmission unit, a processing unit and a sensing unit. In the simulation, all these components are considered. Node class has attributes of; position, sensed data, battery level, sensing capacity, sending capacity and transmission range. There are three types of nodes in the simulation environment. A node can be a normal node, a cluster head node or a sink node. The type property of node class determines the type of that node and a node's type may change during the normal operation (cluster head or sink). Nodes store query table, gathered responses, processed messages, joined nodes, cluster nodes, and path of messages sent. Nodes in the simulation are created at the beginning of the simulation and their locations are set according to the topology simulated (regular or random). Methods of the node classes are summarized in Table 4.1.1.1.

Table 4.1.1.1, Methods of Node Class

Method Name	**Get values**	**Purpose**
FindNeighboors	ArrayList<Node> nodeList	Find nodes in the transmission range
findNeighboorsExceptParent	ArrayList<Node> nodeList, Node parent	In inter cluster communication , finds nodes except parent
messageDelay	Node node	Calculates delays related to the message

Table 4.1.1.1 (continued)

	Node node	Calculates cost of sending a message in node-node communication.
receiveMessageCost	Node node	Calculates cost of receiving a message in node-node communication.
distance	Node node	Calculate Euclidean distance between two nodes.
send	Node aNode, Message message	Sends a message to another node
receive	Node aNode, Message message	Receives a message sent from another node.
ping	Node node	Checks to see if the node to send message is alive or not
pong	Node n	Responds to a ping message
IsProcessed	int messageID	Checks whether the message is processed before or not
markProcessed	int messageID	Marks a message as processed
addNodeToCluster	Node child	Adds a node to the cluster in cluster formation phase

Table 4.1.1.1 (continued)

SetPathtoRoot	ArrayList<Node> newClusterNodes	Adds a new path information to the sink
SetNewClusterNodes	ArrayList<Node> newClusterNodes	Adds new cluster nodes to the list
GetPathToRoot	-	Gets path to sink node

increaseChildCount	Node node	Increases the child count according to the child node responses.
AddQueryToQueryTable	QueryEntry	Adds query entry to the nodes query table
getClusterHead	-	Checks whether the node is a cluster head or not
getBatteryLevel	-	Gets battery level of the node
UpdateBatteryLevel	Double	Updates battery level of the node

4.1.2 Message Class

Message class represents all message types used in the network. There are seven types of messages: advertisement, join, join request, join advertisement, join advertisement request, information messages, node to cluster head message and cluster head to sink message. Methods that are declared in the message class are given in Table 4.1.2. Message size varies according to the message type. In the simulation, instant messages are created according to the conditional changes with message class. All messages contain: message ID, sender, query life time, query, path, depth and path in common.

Table 4.1.2.1, Methods of Message Class

Method Name	**Get values**	**Purpose**
getMessageID	-	Gets the message id
getSender	-	Gets sender of the message
getLifeTime	-	Gets lifetime of the

		query in the message
getQuery	-	Gets query in the message
getPath	-	Gets path specified in the message
getDepth	-	Gets depth specified in the message

4.1.3 Entry Class

Nodes in the simulation respond to the messages by creating tasks that are to be performed after a predefined time period. Nodes process messages and may create some tasks. Those tasks are stored in an entry object. Entries are the tasks that should be processed at a specific time instant. Entry has two main types which are send and receive. Send type entries are related to a sending procedure and receive type entries are related to a receive procedure. Each entry object contains; node that created the entry, nodes send list, time stamp of the entry and message type.

Table 4.1.3.1, Methods of Entry Class

Method Name	**Get values**	**Purpose**
getTimeStamp	-	Gets time stamp of the entry
setTimeStamp	Double	Sets the time stamp in the message
setSender	Node	Sets the sender of the message
getSendList	-	Gets the send list in the message

setSendList	Array List	Sets send list
getNode	-	Gets node that created the entry
setNode	Node	Sets node that created the entry
getMessage	-	Gets message in the entry
setMessage	Message	Sets message in the entry
isProcessed	-	Checks whether the entry is processed or not
setProcessed	-	Sets entry as processed
getType	-	Gets type of the entry
getSender	-	Gets sender in the entry
setBatteryLevel	-	Sets battery level of the sender node
getBatteryLevel	-	Gets the battery level of the sender node

Simulation processes entries according to their types. If the entry type is receive, entry controller invoke receive methods of the related nodes to process message in the entry. Send type entry is processed to find neighboring nodes to create a send list, entries composed from this send list is added to the task list of the main simulation. Messages class defines the methods listed in Table 4.1.3.

4.1.4 GraphicUtil Class

There are several conditions that should be tracked and logged for evaluation purposes. Those conditions are; instant battery consumption of nodes, total battery of nodes, message counts, changes in living node count. Conditions that change in time are processed with methods in graphicsUtil class. Methods in this class are called when events occur in mentioned conditions. Using this class and its method time - battery change, total living nodes - time, total battery change – time, and total messages produced – time graphs are obtained.

Table 4.1.4.1, Methods of graphicUtil Class

Method Name	Get values	Purpose
GetTotalBatteryandTime	Time, node list	Calculates the total battery and its change in time
getTotalBattery	Node list	Gets total battery of nodes
getLivingNodeCount	Time, node list	Gets total living nodes, and shows the change in time

The methods of graphicUtil class are listed in Table 4.1.4.

The flowcharts for the simulation software are presented in Appendix A.

4.2 Performance of evaluation of proposed algorithms

In this section, performances of the clustering algorithms proposed in the Chapter 3 are compared. Those algorithms are:

- Model I: The first node satisfying the query in the advertisement becomes the cluster head.
- Model II: The node closest to the center of gravity becomes the cluster head.
- Model III: The node which has the highest remaining energy becomes the cluster head.

In order to evaluate the performance of these clustering algorithms, the simulation system is initialized with locating a set of nodes in the sensing area and assigning sensed data to each node. We consider two different topologies: regular and random. In the random deployment case, the locations of the nodes are stored in a file in order to test the same deployment in the evaluation of different algorithms. The sensed data in each node is assigned randomly in the range of 0-100 and these values are also saved in a file to use the same set of values in different simulations.

In the simulations interest propagation is started by the sink node that asks for a specific condition. Sink node prepares an advertisement type message that contains query information and broadcasts this message. All nodes in the transmission range of the sink receive this message and process it by comparing their data with the query stated in the message. Nodes do not accept the query if their data doesn't satisfy the condition, the message has been processed before, or their type is cluster head or joined.

Moreover, each node's battery level is decreased with when messages received. The amount of energy spent is determined by the message size and message processing conditions. The energy spent is directly proportional to the message size and message size is determined according to the data that it contains. Nodes also consume energy when they are transmitting messages. Amount of energy spent is determined according to the length of the message and the transmission range. In the simulation environment, transmission range is fixed to a predefined value and nodes are not capable of adaptive power control. Therefore, transmission cost is also proportional to the message length. Moreover, battery levels decrease constantly due to continuous sensing and processing in the nodes. For this purpose, each node's battery level is updated in each simulated millisecond.

The simulation time is determined according to the two predefined values. Round time specify how long inter-cluster and intra-cluster communication phases will take. Round count determines how many rounds the cluster will live. Those variables are specified by the sink in the advertisement message sent and the nodes behave accordingly.

The following data are collected during simulation:

- Energy consumed
- Death time of the first node
- Death time of the first cluster head
- Number of alive nodes at each time instant
- Total number of messages exchanged in the network
- Total number of responses received by the sink
- Total count of failed cluster head to sink messages

In the following simulations, unless otherwise specified, the round time is fixed to 6000 msec, and round count is set to 50, 100, or 150. The simulations are carried out for three cases. Case 1 employs the simplest clustering algorithm in which the first node initiating the cluster becomes the cluster head. In Case 2, the node closest to the center of gravity among the nodes forming the cluster becomes cluster head. In Case 3, the node in the cluster which has the maximum remaining energy becomes the cluster head. For intra cluster communication mode transmission range of the nodes are set to 10 meters and for cluster head to sink communication mode transmission range of the cluster heads are set to 75 meters.

In the simulation, each node can use two different channels. Channel one is used for interest propagation, and each node listens that channel in order to receive interest. In intra cluster communication mode, TDMA Medium Access Control mechanism is employed on channel two and each child node is assigned a time slot for data communication. In intra cluster communication, neighboring clusters use different channels to reduce interference. Cluster heads collect data from the cluster members and after aggregating the data they use channel one for direct communication with the sink node. In order to be connected to the sink node, Cluster heads adjust their transmission power according to 75 meter sending range.

4.2.1 Regular Topology

In regular deployment case, 120 nodes are distributed over the 100x100 sensing region as shown in Figure 4.2.1.1 and the sink is located at the center of the sensing region. In the simulations, each node's battery level is initialized to $9*10^6$ Joules and transmission range for intra cluster communication mode is set to 10 m. The nodes' sensed data are set randomly in the range [0-100]. In the simulations, 24 queries are sent by the sink node. The queries given in Table 4.2.1.1 are repeated in the given order three times to have those 24 queries. Round count is set to 50, 100 or 150. Each round takes 6000 milliseconds. Moreover, round time and round count variables are carried in the advertisement messages which are created by the sink node for interest propagation.

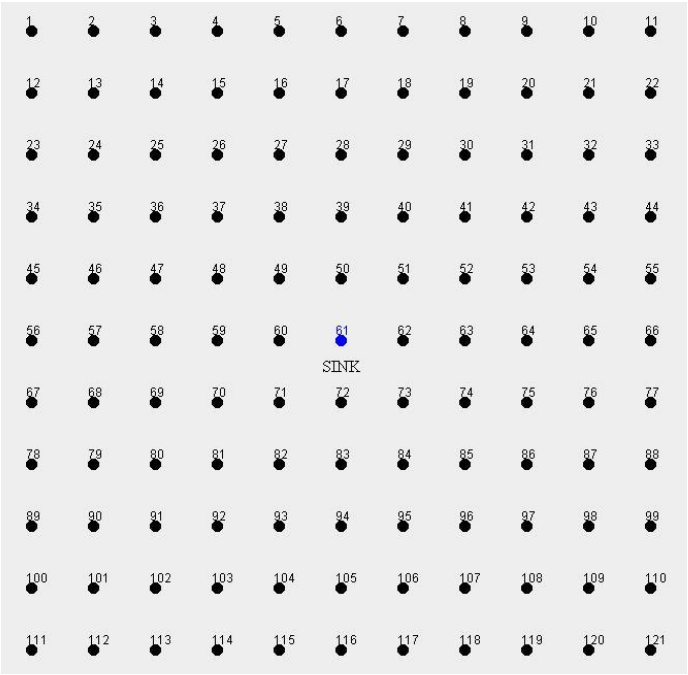

Figure 4.2.1.1: Regular topology simulated

Table 4.2.1.1, Queries and their associated sensor measurement conditions

Query Number	Sensor Data Interval
1	0 – 30
2	70 - 100
3	10 - 40
4	60 - 90
5	30 - 60
6	50 - 80
7	40 - 70
8	20 - 50

Table 4.2.1.2, Results for Round Count=50

<table>
<tr><td colspan="4"><b>Round Count:</b> 50
<b>Round Time:</b> 6000 ms
<b>Transmission Range:</b> 10 m
<b>Query Count:</b> 24
<b>Network Distribution:</b> Regular</td></tr>
<tr><th>Algorithms</th><th>Energy Consumed</th><th>Minimum Remaining Energy of Nodes</th><th>Maximum Remaining Energy of Nodes</th></tr>
<tr><td><b>Model I</b></td><td>189494872</td><td>6450394</td><td>8148534</td></tr>
<tr><td><b>Model II</b></td><td>189709715</td><td>6478766</td><td>8146221</td></tr>
<tr><td><b>Model III</b></td><td>190057924</td><td>6851407</td><td>8003535</td></tr>
</table>

In Table 4.2.1.2, consumed energy, minimum remaining energy of nodes after simulation and maximum remaining energy of nodes at the end of the simulation are given for round count 50. According to the consumed energy, Model III that selects the node that has the highest battery level as the cluster head consumes more energy compared to Model I and Model II. According to consumed energy, the models are ordered as Model III > Model II > Model I as shown in Figure 4.2.1.5. The worse performance of Model II and Model III are mainly caused by the extra energy consumed during cluster head reselection phase in these models. As the results indicate, according to minimum remaining energy of nodes, models are ordered as Model III > Model II > Model I. If we define the lifetime of the network as the time elapsed until the first node dies, this result suggests that the lifetime of the network will be highest in Model III and lifetime with Model II is longer than lifetime of Model I. Therefore, from the lifetime point of view Model III is the best among three models, as it chooses the highest energy node as the new cluster head at the beginning of the intra-cluster communication.

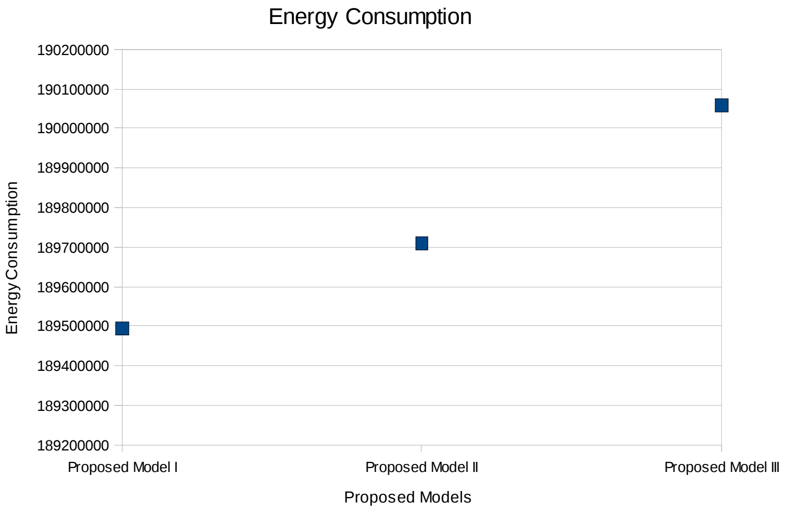

Figure 4.2.1.5: Comparison of Energy Consumption for Round Count=50

Table 4.2.1.3, Results for Round Count=100

Round Count: 100 **Round Time:** 6000 ms **Transmission Range:** 10 m **Query Count:** 24 **Network Distribution:** Regular			
Algorithms	**Energy Consumed**	**Minimum Remaining Energy of Nodes**	**Maximum Remaining Energy of Nodes**
Model I	367423307	4041661	7364408

Model II	367423665	4070122	7362095
Model III	367941995	4801653	7099939

In Table 4.2.1.3, consumed energy, minimum remaining energy of nodes after simulation and maximum remaining energy of nodes at the end of the simulation are given for round count 100. According to consumed energy, the models are ordered as Model III > Model II > Model I as shown in Figure 4.2.1.6. As in round count=50 case, Model III consumes more energy compared to Model I and Model II. However, in this case, energy consumed by Model II is approximately equal to energy consumed by Model I and difference between energy consumption levels of Model I, Model II and Model III reduces. The main reason for this is that energy consumed during cluster head reselection phase is amortized by lower energy consumption during rounds. As the results indicate, according to the minimum remaining energy of nodes, models are ordered as Model III > Model II > Model I. This result also suggests that the lifetime of the network will be highest in Model III and lifetime with Model II is longer than lifetime of Model I and difference between lifetimes will be much higher compared to round count=50 case.

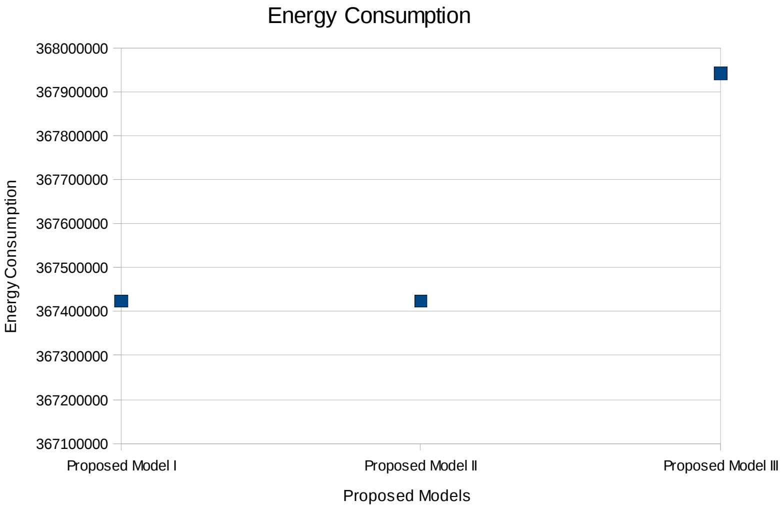

Figure 4.2.1.6: Comparison of Energy Consumption for Round Count=100

In Table 4.2.1.4, consumed energy, minimum remaining energy of nodes after simulation and maximum remaining energy of nodes at the end of the simulation are given for round count 150. According to consumed energy, the models are ordered as Model III > Model I > Model II as shown in Figure 4.2.1.7. Like round count 50 and 100 cases, according to the consumed energy, Model III consumes more energy compared to Model I and Model II. However, in this case, the energy consumed by Model II is lower than the energy consumed by Model I. The main reason for this is that the extra energy consumed during cluster head reselection phase is much lower than the gain obtained due to lower energy consumption for intra cluster communication in Model II. As the results indicate, according to minimum remaining energy of nodes, models are again ordered as Model III > Model II > Model I and difference between lifetimes will be much higher compared to round count 50 and 100 cases. As the results indicate, with higher round counts, in order to reduce energy consumed per query, Model II must be chosen and in order to increase the lifetime of the network, Model III must be chosen.

Table 4.2.1.8 presents battery consumption figure for the first four queries for round count 150. In the simulation, energy consumed during cluster formation, intra cluster communication and cluster head to sink communication phases are measured. In Model I, as the re-clustering is not done, re-clustering phase energy consumption is always 0. In all models, cluster heads use a separate channel with higher transmission power for cluster head to sink communication in a single hop. For cluster head to sink communication cluster heads adjust their transmitters to 75 m range to reach the sink node in single hop and since the transmission power is much higher, large amount of energy is spent for cluster head to sink communication. As the same configuration (topology and sensed data) are used in all simulations, the same set of clusters are formed in all Models. Therefore, battery consumption of cluster head to sink communication is the same for all Models.

Table 4.2.1.4, Results for Round Count=150

Round Count: 150 **Round Time:** 6000 ms **Transmission Range:** 10 m **Query Count:** 24 **Network Distribution:** Regular			
Algorithms	**Energy Consumed**	**Minimum Remaining Energy of Nodes**	**Maximum Remaining Energy of Nodes**
Model I	545351743	1632928	6580283
Model II	545137586	1661479	6577969
Model III	545819355	2751900	6196343

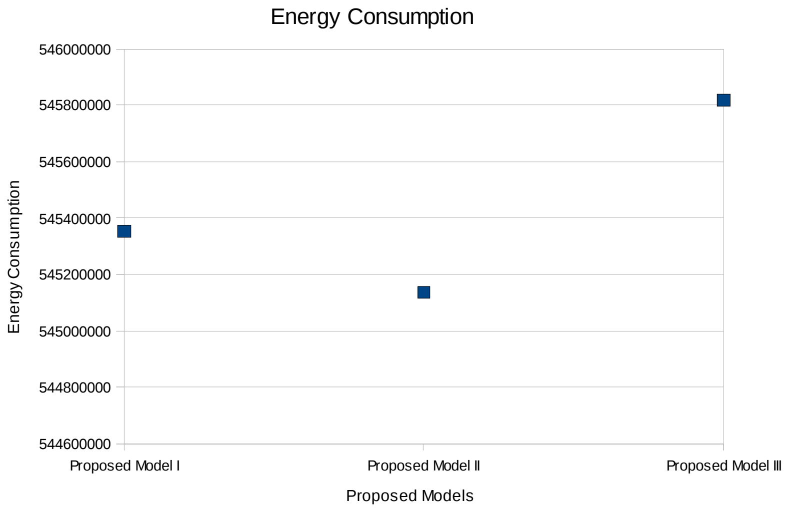

Figure 4.2.1.7: Comparison of Energy Consumption for Round Count=150

Table 4.2.1.8, Energy Consumption for the First Four Queries

Query 1 - (0-30)			
	Proposed Model I	**Proposed Model II**	**Proposed Model III**
Re-Clustering Battery Consumption	0	28375	27149
Intra Cluster Communication Battery Consumption	1082801	1031726	1113446
CH to Sink Communication Battery Consumption	9789392	9789392	9789392
Query 2 - (70-100)			
	Proposed Model I	**Proposed Model II**	**Proposed Model III**
Re-Clustering Battery Consumption	0	41314	46308
Intra Cluster Communication Battery Consumption	1631012	1549292	1631012
CH to Sink Communication Battery Consumption	20430036	20430036	20430036
Query 3 - (10-40)			

	Proposed Model I	Proposed Model II	Proposed Model III
Re-Clustering Battery Consumption	0	63288	77362
Intra Cluster Communication Battery Consumption	2608258	2475463	2577613
CH to Sink Communication Battery Consumption	29793802	29793802	29793802
Query 4 - (60-90)			
	Proposed Model I	**Proposed Model II**	**Proposed Model III**
Re-Clustering Battery Consumption	0	72459	77362
Intra Cluster Communication Battery Consumption	3187114	3013459	3156469
CH to Sink Communication Battery Consumption	39583194	39583194	39583194

As it can be seen from Table 4.2.1.8, energy consumed during intra cluster communication is the lowest in Model II. According to this observation, we can conclude that, as the round count increases, the energy consumed during the cluster re-selection will be paid back during intra cluster communication and this explains why Model II performs better as the round count increases. The main source of savings in the intra-cluster communication phase is the re-selection of cluster head before intra-cluster communication starts. In Model I, the node closest to the sink receives the interest message first and becomes the cluster head. However, in this case, the average number of hops that the cluster members reach cluster head increases as shown in Figure 4.2.1.8. The same cluster formed according to Model II is shown in Figure 4.2.1.9. As it can be seen from the figure, Model II selects the node closest to the center of gravity of the nodes in the cluster. This results in decreased average number of hops to reach cluster head. Thus, energy consumed during intra cluster communication is lower compared to Model I.

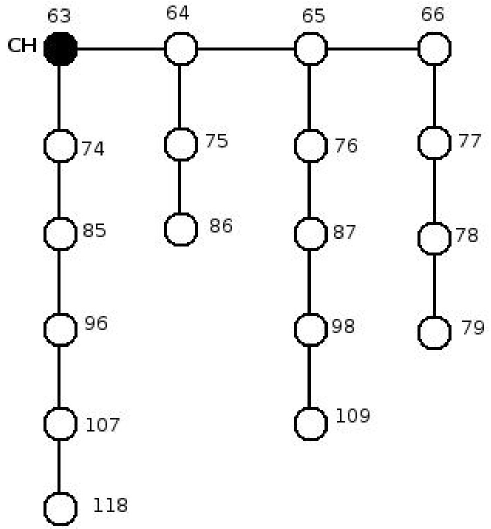

Figure 4.2.1.8: A Cluster Formed according to Model I

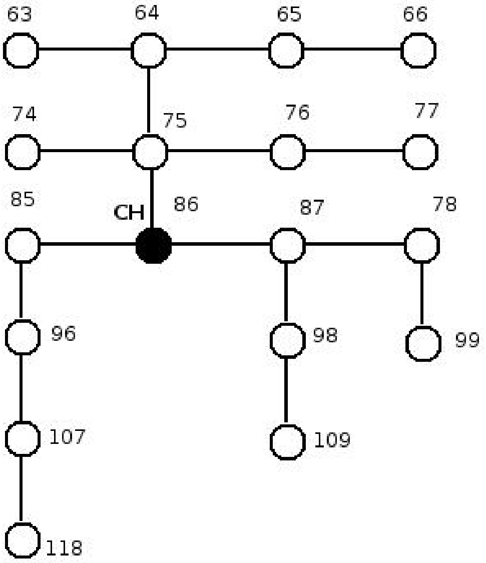

Figure 4.2.1.9: A Cluster Formed According to Model II

4.2.2 Random Topology

In order to evaluate the performance of proposed clustering models on random topologies 250 nodes are deployed randomly over a 100x100 sensing region. The topology employed is given in Figure 4.2.2.1. As in the regular deployment case, the sensed data for each node is also randomly assigned in the range 0-100. In order to compare different cases in equal conditions, locations of the nodes and sensed data stored in a file and the same configuration is used in all simulations. Therefore it is guaranteed that the proposed algorithms are evaluated under the same conditions.

In the simulations, the sink node is placed at the center of the sensing area, initial battery level of each node is set to $9*10^6$ Joules, transmission range for intra-cluster communication is set to 10m, transmission range for cluster hea**d** to sink communication is set to 75m and three sets of simulations are performed for round counts 50, 150 and 300. In each simulation, 8 queries given in 4.2.1.1 are sent by the sink node and energy consumption for these queries for the simulated clustering model is measured.

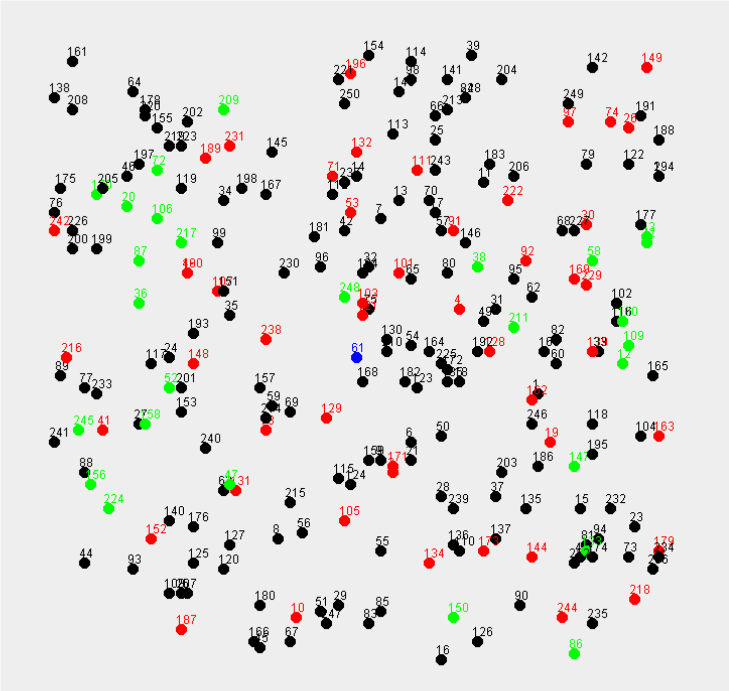

Figure 4.2.2.1: Randomly Deployed Sensor Network

Table 4.2.2.1, Results for Round Count=50

Round Count: 50

Round Time: 6000 ms

Transmission Range: 10 m

Query Count: 8 **Network Distribution:** Random **Node Count:** 250			
Algorithms	**Energy Consumed**	**Minimum Remaining Energy of Nodes**	**Maximum Remaining Energy of Nodes**
Model I	148743271	8119720	8762819
Model II	149090932	8121620	8762774
Model III	149278702	8121621	8762775

In Table 4.2.2.1, consumed energy, minimum remaining energy of nodes after simulation and maximum remaining energy of nodes at the end of the simulation are given for round count 50. As in the regular topology case, according to the consumed energy, Model III consumes more energy compared to Model I and Model II in random topology. According to consumed energy, the models are ordered as Model III > Model II > Model I as shown in Figure 4.2.2.1. Again, according to minimum remaining energy of nodes, models are ordered as Model III > Model II > Model I, but minimum remaining energy of nodes in Model III and Model II are very close to each other. This result reveals that the lifetime of the network will be highest in Model III and lifetime with Model II and both lifetimes are longer than the lifetime of Model I. Therefore, from the lifetime point of view Model II and Model III is the best among three models.

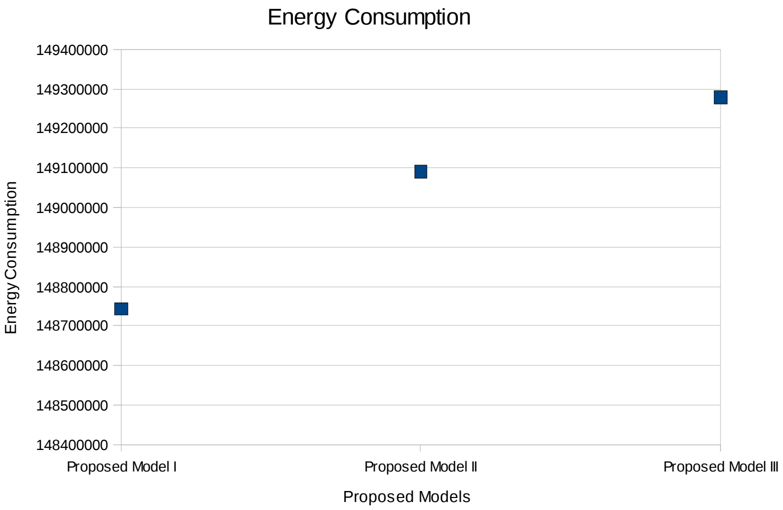

Figure 4.2.2.1: Comparison of Energy Consumption for Round Count=50

In Table 4.2.2.1, consumed energy, minimum remaining energy of nodes after simulation and maximum remaining energy of nodes for the random topology are given for round count 150. As in the regular topology case, according to the consumed energy, in random topology, Model II is the best as it consumes less energy compared to Model I and Model III. According to consumed energy, the models are ordered as Model III > Model I > Model II as shown in Figure 4.2.2.2. According to minimum remaining energy of nodes, models are ordered as Model III > Model II > Model I, but as in the round count=50 case, minimum remaining energy of nodes in Model III and Model II are very close to each other. Therefore, lifetime of the network will be high in Model III and Model II as cluster heads chosen by these models lead to better energy consumption than Model I. Therefore, from the total energy point of view Model II is the best and from the lifetime point of view Model II and Model III are better than Model I.

Table 4.2.2.2, Results for Round Count=150

Round Count: 150 **Round Time:** 6000 ms **Transmission Range:** 10 m **Query Count:** 8 **Network Distribution:** Random **Node Count:** 250			
Algorithms	**Energy Consumed**	**Minimum Remaining Energy of Nodes**	**Maximum Remaining Energy of Nodes**
Model I	394132232	6513898	8282819
Model II	394050863	6516618	8282774
Model III	394495367	6516621	8282777

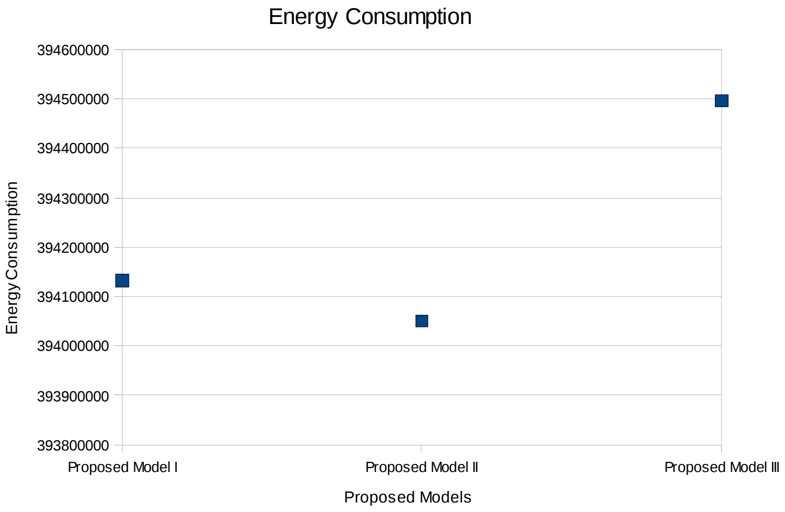

Figure 4.2.2.2: Comparison of Energy Consumption for Round Count=150

In Table 4.2.2.3, consumed energy, minimum remaining energy of nodes after simulation and maximum remaining energy of nodes at the end of the simulation are given for round count 300. According to consumed energy, the models are ordered as Model III > Model I > Model II as shown in Figure 4.2.2.3. Similar to round count 50 and 150 cases, Model III consumes more energy compared to Model I and Model II and the energy consumed by Model II is lower than the energy consumed by Model I. As the results indicate, according to minimum remaining energy of nodes, models are again ordered as Model III > Model II > Model I. As the results indicate, likewise the regular topology case, in order to reduce energy consumed per query, Model II must be chosen and, on the other hand, in order to increase the lifetime of the network, Model III must be chosen.

Table 4.2.2.3, Results for Round Count=300

Round Count: 300

Round Time: 6000 ms

Transmission Range: 10 m

Query Count: 8

Network Distribution: Random

Node Count: 250			
Algorithms	**Energy Consumed**	**Minimum Remaining Energy of Nodes**	**Maximum Remaining Energy of Nodes**
Model I	762215675	4105165	7562819
Model II	761490761	4109115	7562774
Model III	762235547	4109120	7562779

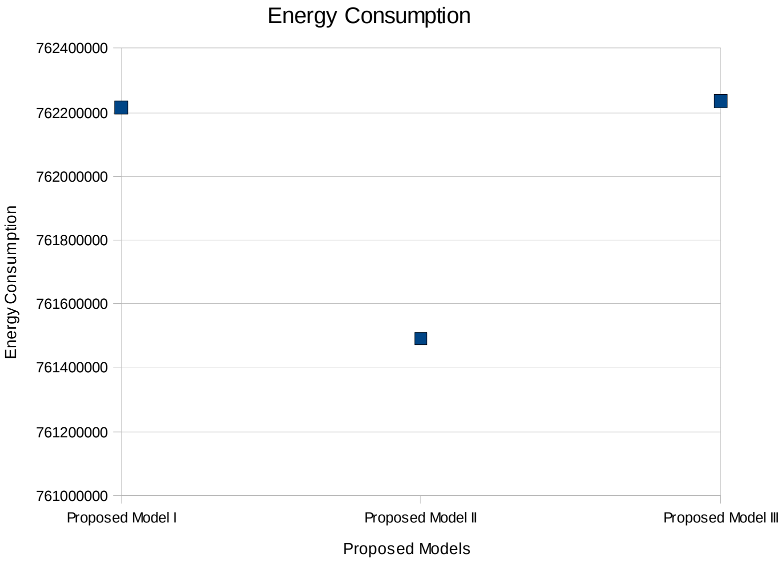

Figure 4.2.2.3: Comparison of Energy Consumption for Round Count=300

Comparing the Figure 4.2.2.1, Figure 4.2.2.2 and Figure 4.2.2.3 it can be inferred that when round count is increased selecting the node that is close to the center of area covered by the cluster as the cluster head performs better. Therefore, Model II leads to less energy consumption for intra cluster communication and if round count variable increases the energy gain increases and the energy consumed during cluster head re-selection, which is the case in Model II and Model III, is amortized by the lower energy consumption during rounds. From the lifetime point of view, as the Model III distributes energy depletion evenly in the network, the minimum remaining energy of nodes is higher compared to other two models. This leads to longer lifetime with Model III.

CHAPTER 5

CONCLUSION

In this thesis study, the clustering algorithm proposed in [1] is discussed and two new energy efficient clustering algorithms based on this algorithm are proposed. Algorithm proposed in [1] is a clustering algorithm with semantic properties. In this algorithm there are three types of messages that are used for query dissemination, cluster formation and cluster maintenance phases. Sink node sends advertisement message to the network through neighboring nodes in its transmission range. The first node that satisfies the query declares itself as the cluster head and starts cluster formation. We call this clustering algorithm Model I.

In this thesis, we add a new re-clustering phase after completion of cluster formation and before data gathering phases of Model I to improve energy efficiency. We propose two such algorithms. In Model II, the node that is closest to the center of the area covered by the cluster members is chosen as the new cluster head. In Model III, the node that has the highest remaining energy is chosen as the new cluster head.

In order to, prove and compare efficiency of proposed algorithms, a computer simulation software has been developed. The simulator has been developed in Java programming language. The algorithms considered have been evaluated over regular and random topologies on 100x100 sensing region. In regular topology, 120 nodes are located in a mesh manner with 10m spacing between the neighboring nodes. In random topology, 250 nodes are randomly placed in the sensing area. In both topologies, the sink is located at the center of the sensing area and the sensor measurements are assigned to nodes randomly. In order to evaluate the energy efficiency under different conditions, for regular and random topologies, the round count is set to 50,100,150 and 300 with round time 6000 msec, intra cluster communication mode transmission range is set to 10 m, and cluster head to sink mode transmission range is set to 75 m. For the simulations, locations and sensor measurements of the nodes are saved in a file and a predefined set of queries are applied during simulation. Therefore, all clustering models are evaluated under the same conditions.

the area covered by the cluster is chosen as the cluster head. With such a cluster head selection method, the average hop count for child nodes to reach and deliver their messages to the cluster head decreases and this leads to reduced energy consumption for intra cluster communication phase. The energy consumption for transmitting a message is directly proportional to length of the message. Each parent node receiving data from its children aggregates those messages and send aggregate to its parent node. Therefore, increase in the hop count also increases message sizes and so energy consumption increases.

In model I, the node that receives the advertisement message sent by the sink node first is selected as the cluster head. In model III, the node that has the highest battery level in the cluster is selected as the cluster head. That is, the node that is away from the sink node or close to the edge of the area covered by the cluster is likely to be new cluster head. Therefore, model I and model II leads to longer paths to cluster head within the cluster. Therefore, with model I and model II, energy consumption is higher for intra cluster communication as compared to model II.

In model II and model III, there is an extra energy cost caused by the introduced cluster head re-selection phase. As the simulation results show, this extra cost is compensated by the gains obtained during intra cluster communication phase. According to simulation results, with small round counts, model I consumes less energy compared to other models, however, model II becomes better as the round count increases.

Results of the simulations also point out that model III has the highest energy consumption in all cases compared to other models. However, if we look at the minimum remaining energy across the nodes, model III has the largest minimum remaining energy. If we define lifetime of the network as the time elapses until the first node runs out of energy, minimum remaining energy implies the lifetime of the network. Therefore, as the simulation results show, from the lifetime point of view, model III is the best clustering method and model I is the worst clustering method.

As a summary, as the round count increases, the total energy consumed is the smallest with model II. However, for smaller round counts, model I is the best as there is no cluster head re-selection overhead. On the other hand, if longer lifetime is needed, model III is the best clustering method among these three methods.

In this thesis, we assumed that cluster heads send aggregate data to sink directly over a separate channel in one hop. We also assumed that the nodes adjust their transmitters according to fixed ranges for intra cluster and cluster head to sink communication modes. As a future work, adaptive power control can be introduced into the communication models, and the effects of adaptive power control on the performances of proposed models can be evaluated. In addition, instead of using single hop communication between cluster heads and the sink, multi hop communication alternative could also be investigated. Another future work could be to propose another re-clustering method that takes into account the existing network topology and performing global optimization to reduce energy consumption.

REFERENCES

[1] M. Merabti, and H. Mokhtar , A Semantic Clustering Routing Protocol for Wireless Sensor NetworksF.Bouhafs,. IEEE CCNC 2006 Proceedings, pp. 351-355.

[2] F.Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci , A Survey on Sensor Networks, , IEEE CCNC 2002 Proceedings, pp. 102-114.

[3] Kemel Akkaya, Mohamed Younis A survey on routing protocols for wireless sensor networks, , Science Direct 2003,pp. 325-349.

[4] Joanna Kulik, Wendi Rabiner, and Hari Balakrishnan, Adaptive Protocols for Information Dissemination in Wireless Sensor Networks, MIT, 2006, ACM/IEEE, pp.135-142.

[5] David B. Johnson, David A. Maltz, Josh Broch DSR: The Dynamic Source Routing protocol for Multi-Hop Wireless Ad Hoc Networks, Carnegie Mellon University, In Ad hoc networking (December 2000), pp. 139-172.

[6] Nicklas Beijar, Zone Routing Protocol (ZRP), , Helsinki University of Technology

[7] , Jim Snow, Wu-chi Feng, Wu-chang Feng, , Implementing a Low Power TDMA Protocol Over 802.1, 1Portland State University, Wireless Communications and Networking Conference, 2005 IEEE , pp. 75-80.

[8] Lisa DiPippo, David Tucker, Victor Fay-Wolfe, Kevin L Bryan, Tiegeng Ren, Wiliam Day, Matthew Murphy, Tim Henry, Shaun Josep, Energy-efficient MAC for Broadcast Problems in Wireless Sensor Networks, in Proceedings of the Third International Conference on Networked Sensing Systems 2006

[9] Wei Wang, Honggang Wang, Dongming Peng, Hamid Sharif, An energy Efficient Pre-Schedule Scheme for Hybrid CSMA/TDMA MAC in Wireless Sensor Networks, , University of Nebraska-Lincoln IEEE 2006

[10] Anahit Martirosyan, Azzzedine Boukerche and Richard W. Nelem Pazzi , A Taxonomy of Cluster-based Routing Protocols for Wireless Sensor Networks, , IEEE 2008

[11] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan, Energy-Efficient Communication Protocol for Wireless Micro Sensor Networks, , MIT 2000

[12] Min-Woo Park, Jin-Young Choi, Young-Ju Han, and Tai-Myoung Chung, An Energy Efficient Concentric Clustering Scheme in Wireless Sensor Networks, , 2009 Fifth International Joint Conference on INC, IMS and IDC

[13] Choon-Sung Nam, Hee-Jin Jeong and Dong-Ryeol Shin, the Adaptive Cluster Head Selection in Wireless Sensor Networks, 2008 IEEE

[14] Yahya M. Tashtoush and Mohammed A. Okour, Fuzzy Self-Clustering for Wireless Sensor Networks, IEEE 2008

[15] Fei ye, Yao Hua, Zhisheng Niu, Sub Cluster Aided Data Collection in Multihop Wireless Sensor Networks, IEEE 2007

[16] Hamidreza Alipour, Maghsoud Abbaspour, Mostafa Esmaeili, Hamed Mousavi, Hamed Shahhoseini ,DACA: Dynamic Advanced Clustering Algorithm for Sensor Networks, , IEEE 2007

[17] Ameer Ahmed Abbasi, Mohamed Younis , A survey on clustering algorithms for wireless sensor networks, 2007 Computer Communications

[18] F.Bouhafs, M. Merabti, and H. Mokhtar, A Semantic Clustering Protocol for Wireless Sensor Networks, 2006 IEEE

[19] Azzedine Boukerche, Richard Werner N. Pazzi, Regina B. Araujo, HPEQ- A Hierarchical Periodic, Event-driven and Query-based Wireless Sensor Network Protocol, 2005 IEEE

[20] Yun Hu, Fengqi Yu, Ping Deng , A Novel Data Query Method for Wireless Sensor Networks, 2007 IEEE

[21] Muhammad Umer, lars Kulik, Egemen Tahin, Optimizing Query Processing using Selectivity-Awareness in Wireless Sensor Networks, 2009 Computers, Environment and Urban Systems

[22] Octav Chipara, Chenyang Lu, Gruia-Catalin Roman, Real-time Query Scheduling for Wireless Sensor Networks, 2007

[23] Anthony D. Wood, Leo Selavo, John A. Stankovic, SenQ: An Embedded Query System for Streaming Data in Heterogeneous Interactive Wireless Sensor Networks,

[24] Yingjie He, Alan Tully, Query Processing for Mobile Wireless Sensor Networks: State-of-the-art and Research Challenges, 2008 IEEE

[25] Christian HOAREAU, Query language for Location-Base Services: A Model Checking Approach, , 2008 IEICE

[26] Ryan E. Blace , Mohamed Eltoweissy and Wael Abd-Almageed, Threat Aware Clustering in Wireless Sensor Networks,

[27] D.Culler,D. E. M. Srivasta va, “Overview of Sensor Network”, IEEE Computer Magazine, vol. 37, no. 8, pp. 6-28, 2004

[28] S. Tilak et al., A taxonumy of wireless microsensor netwrok models, Mobile computing and communications Review 6 (2) (2002) 28-36

[29] Chang et al., “Maximum Lifetime Routing in Wireless Sensor Networks”, IEEE/ACM Transactions on networking, vol 12 No.4 (2004)[30] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless sensor networks, in: Proceeding of Hawaii, January 2000.

[31] C.Sevgi (2009), Network Dimensioning in Randomly Deployed Wireless Sensor Networks, published doctoral dissertation, Middle East technical University, Department of Informatics Institute, Ankara

APPENDICIES

A. FLOWCHART OF SIMULATION

In the development of the proposed algorithms, the algorithm which was used in [1] is taken as the base algorithm. Proposed algorithms make changes in cluster formation phases of the base algorithm. It offers new cluster formation with selecting new cluster head by considering two conditions. Those conditions are selecting the node which has the highest battery level and selecting the node which is closer to the average coordinates of the cluster. In developing the simulation five fundamental classes introduced in the previous sections are used. The main flow of the simulation is coded in the main method of the simulation. The algorithm consists of main structure, interest propagation, cluster formation, generating schedules and cluster maintenance. The parts of the algorithms are described in the following sections.

A.1 Main Structure

Major activities in simulation are performed in the order defined by the main structure. These activities are initialization of objects, interest propagation, cluster formation and cluster maintenance.

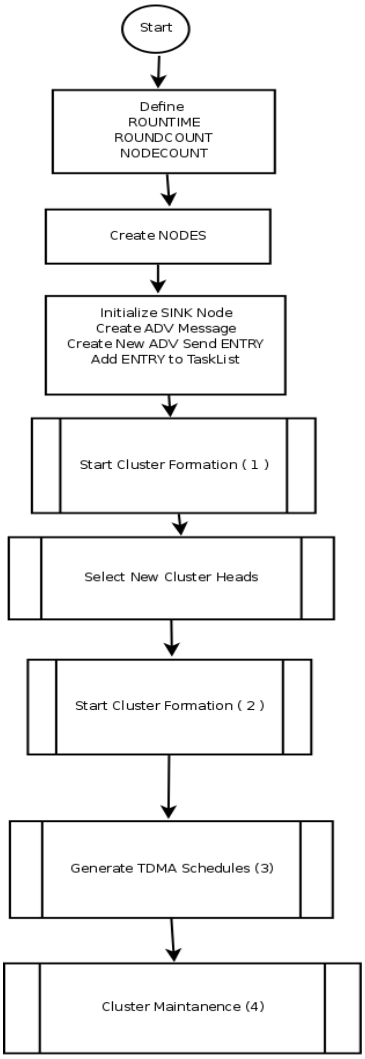

Figure A.1.1 Main flow of simulation

Figure A.1.1 show the order and methods that are called. In the first part, constants of the simulation are declared and initialized. Those constants are ROUNDTIME, ROUNDCOUNT and NODECOUNT. ROUNDTIME is a constant that represents a round time which is used in inter cluster communication. Cluster Head node assigns TDMA schedules to child nodes considering the ROUNDTIME. ROUNDCOUNT describes how many counts will occur in query lifetime.

A query is processed up to ROUNDTIME x ROUNDCOUNT total time. After this time period, clusters are destroyed and nodes set their types to normal. In addition to the constants variables, arrays and array lists that will be used in the simulation are declared. Node count constant, time line array list, node list array list, round log array list are declared. Node count constant represent the number of nodes that will be used in simulation. Time line is the holder of entries created by nodes and it is in type of object array list. Node list is the holder of all nodes created for simulation and it is also in type of object array list. Round log string type array list is used for messaging logs that is produced in cluster formations and cluster maintenance.

Nodes are created after these declarations. The number of Nodes to be created is determined by the number of the nodes declared in initialization part. Node locations are randomly selected in a predefined region of x=0-100,y=0-100. They are assigned a value between 0-100 as the sensing information. Each node except the sink node is in type normal node at the initialization. Nodes' sensed values do not change in time; they are kept as the same during the query processing phase.

Sink node is initialized separately from other nodes. Sink node is located at a specific location of (X=0,Y=0). Sink node has unlimited battery supply and it creates the first advertisement message which will result in cluster formations.

A.2 Initialization Phase

After initialization of nodes and simulation environment, sink node creates the first advertisement message which consists of message ID, sender, query, send list. After producing advertisement message, sink node creates the first entry which is located at the first place of the time line with time stamp 0.0.

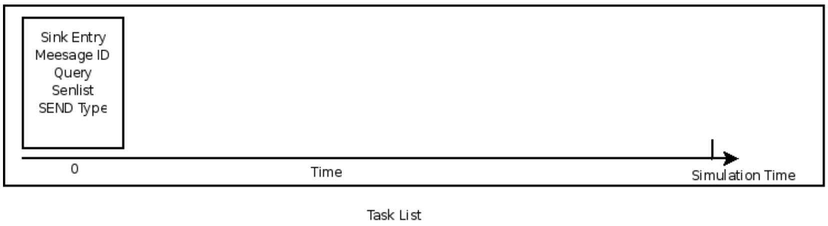

Figure A.2.1 Task list and first entry created by sink node

This entry consists of sender node, list of nodes that message is sent to, message and time stamp. The first entry's type is set as send type. This is the first entry to be processed in starting phase of simulation. Task list is the container of the entries created by nodes and entries are ordered in time of creation (Figure A.2.1).

A.3 Interest Propagation and Cluster Formation

There is a method runs in cluster formation which checks for the new or unprocessed entries. If it detects an unprocessed entry, it extracts the entry and does the tasks that are stated in entry.

This procedure starts after the initialization phase and interest propagation phase. Interest is propagated through the network after the first implementation of the entry created by the sink node. The controlling process implements the first entry and after implementation new entries resulting from the first entry is added to the task list.

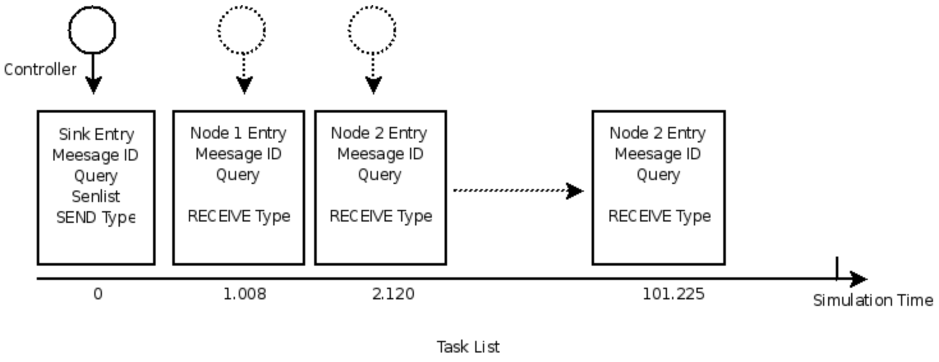

Figure A.3.1 Task list and controller process

The entry is set as processed after the controller process the tasks stated in the entry. Entries, except the first entry created by sink node, are created dynamically and they are added to the task list. The first entry corresponds to a send operation. Controller process finds the nodes in range and make node to transmit the message in range. The transmission cost (i.e., energy) is computed at sink node. Controller checks the available nodes that can receive the message by considering the range of the sink node with calculating Euclidean distance between sink and receiving node. Controller adds received entries of the neighboring nodes of the sink node with adding processing and receiving delay to the time stamp. This process continues until the end of the simulation. Entries that are assigned as processed are ignored. Controlling process always checks for new entries and it process new entries instantly. Node's battery levels are decreased in each send and receive operations. Node types are determined considering the result of the processed messages (Figure A.3.1). The first entry created by the sink node contains the advertisement message. After processing the first entry and adding the resulting receive entries message propagation starts.

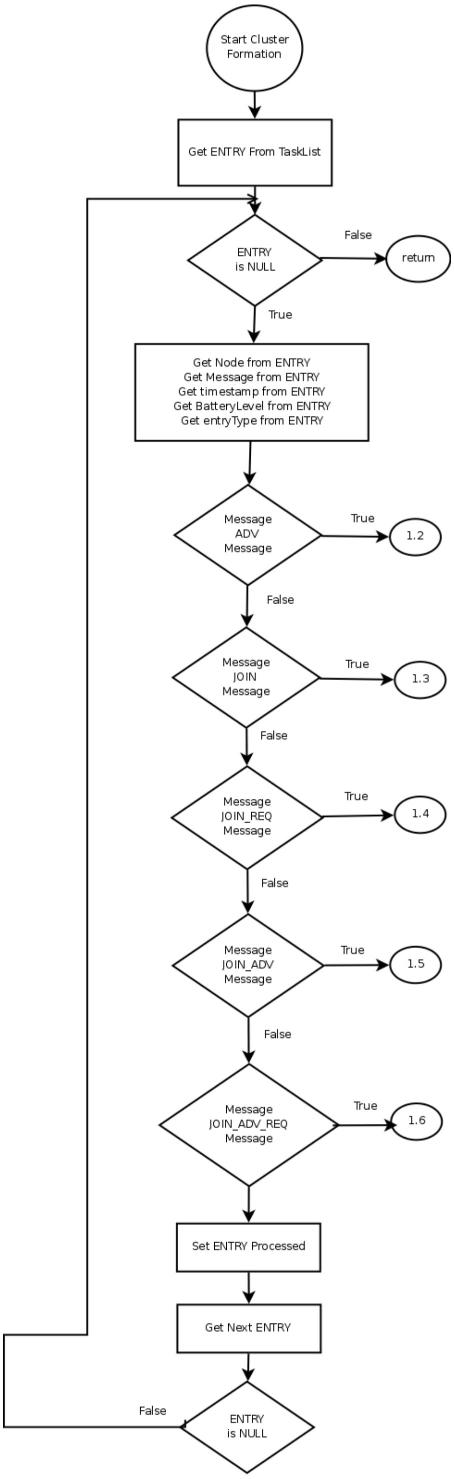

Figure A.3.2 Task list and controller process

The process of processing entries is given in figure A.3.2 Controlling process starts after the initialization of the simulation constants and objects. Controlling process gets the unprocessed entry from the task list.

It extracts node, message, time stamp, battery level and entry type from the entry. Message type is important for processing entries. If the message type is advertisement message, flow is directed to the advertisement message processing.

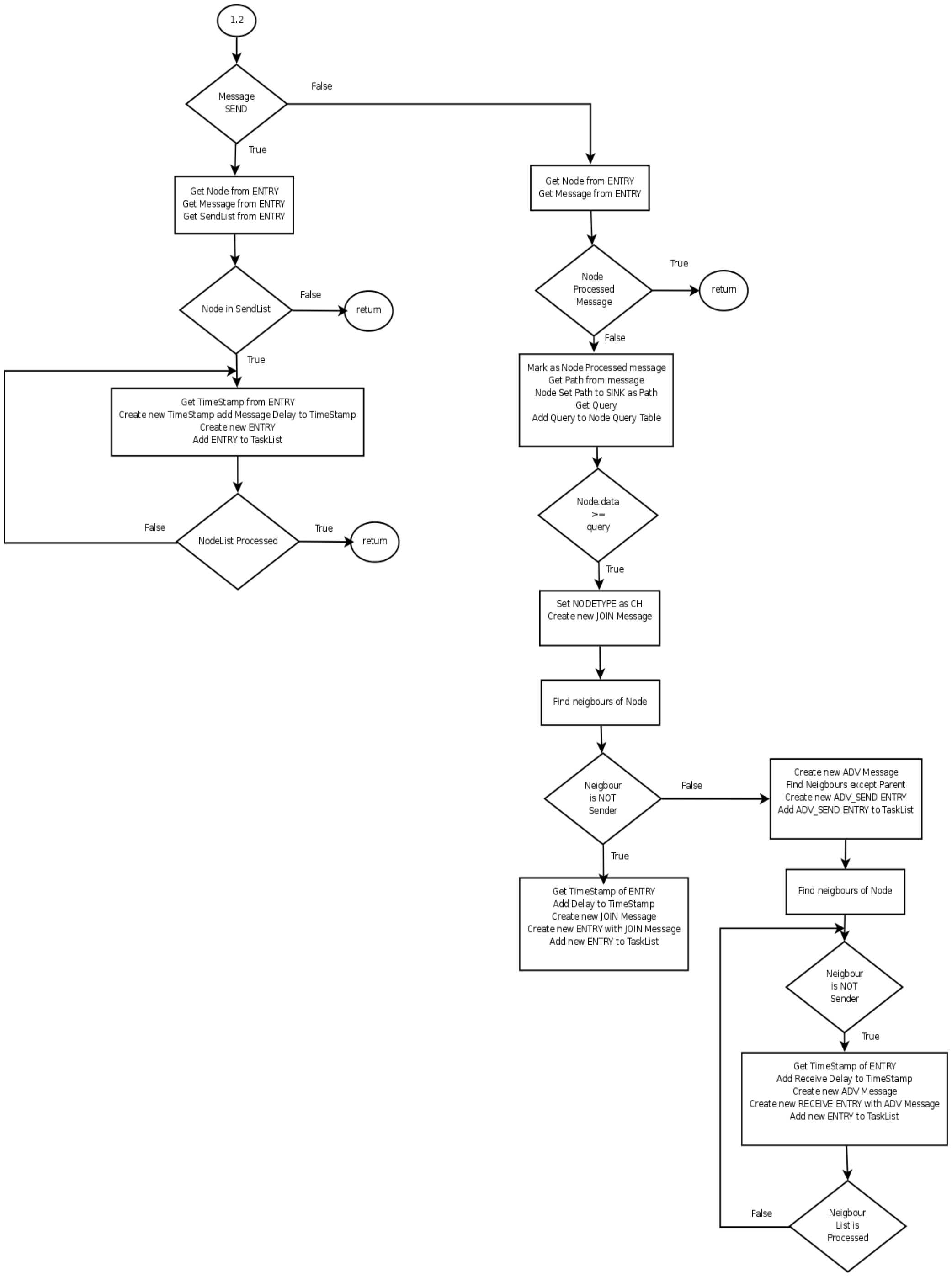

Figure A.3.3 Advertisement message processing

It is checked whether the type of the message is advertisement and the method is send or receive. If the message type is send, sender, message and send list are extracted from the entry. The neighboring nodes are found, and the delay of sending message is added to the entry time stamp. For each neighboring node a receive entry is created and added to the task list with updated time stamp. Delay and order of new entries are determined considering process delay, sending delay with variable Euclidean distance between sending node and receiving node. The process continues until the send list of the send type entry is processed.

If the entry type is receive; sender node and message is extracted from the entry. If the message is processed by the node, the message is not considered. Otherwise, node sets message as processed, path of the message is extracted, query is taken, and query is added to the node's query table. Node's data is compared with the query, if node's data is equal or bigger than the value stated in query, Node's type is set as cluster head and a new join message is composed. The new join message is added to the entries of the receiving neighboring nodes and those entries are added to the time line with delay order. If Node's data does not meet the query, an updated advertisement message is composed with adding node to the path variable in the message. New receive entries are created for neighboring nodes and those entries are added to the task list (Figure A.3.4).

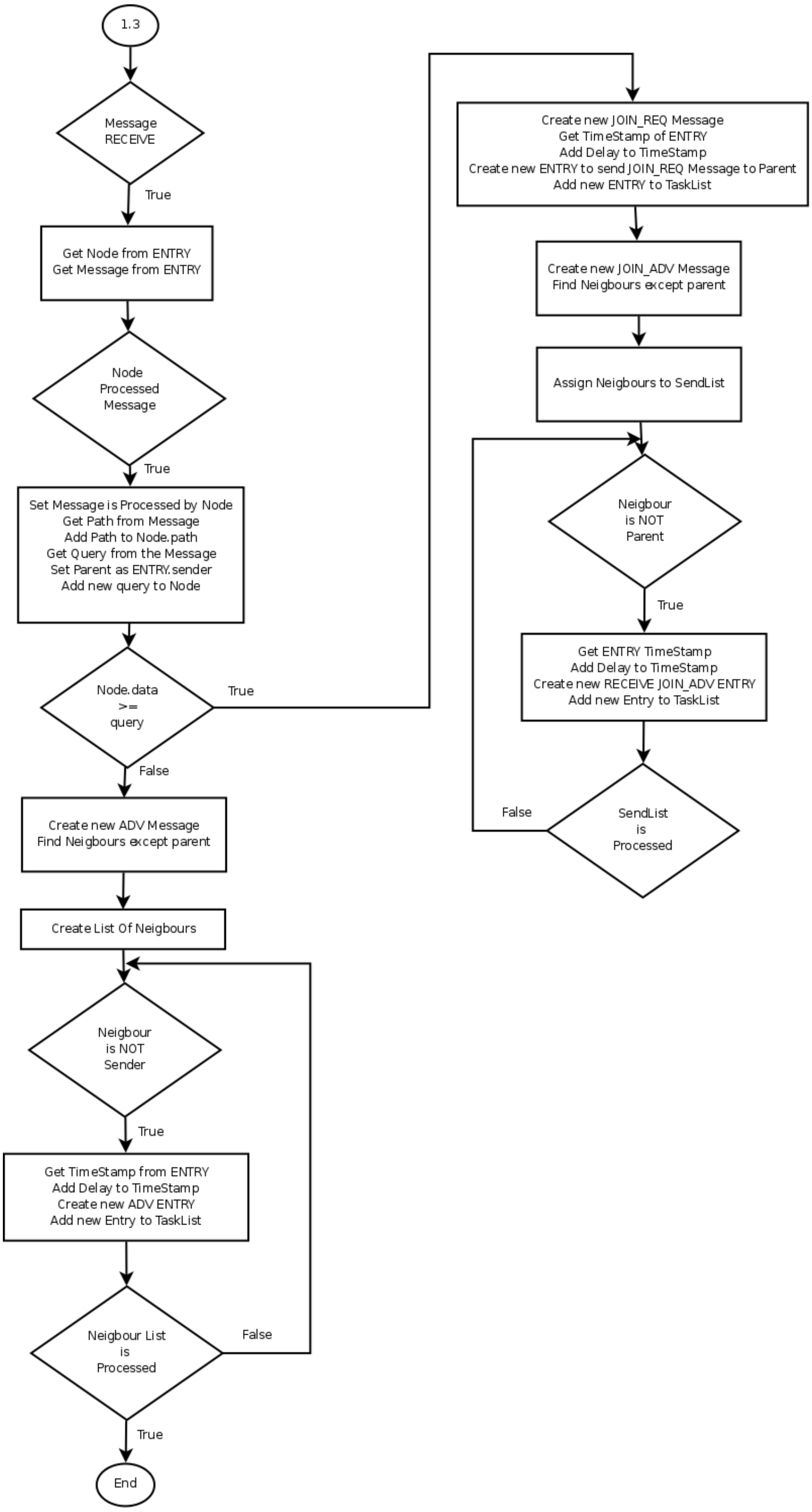
1.3
Message RECEIVE
True
Get Node from ENTRY
Get Message from ENTRY
Node Processed Message
True
Set Message is Processed by Node
Get Path from Message
Add Path to Node.path
Get Query from the Message
Set Parent as ENTRY.sender
Add new query to Node
Node.data >= query
True
False
Create new ADV Message
Find Neigbours except parent
Create List Of Neigbours
Neigbour is NOT Sender
True
Get TimeStamp from ENTRY
Add Delay to TimeStamp
Create new ADV ENTRY
Add new Entry to TaskList
Neigbour List is Processed
False
True
End
Create new JOIN_REQ Message
Get TimeStamp of ENTRY
Add Delay to TimeStamp
Create new ENTRY to send JOIN_REQ Message to Parent
Add new ENTRY to TaskList
Create new JOIN_ADV Message
Find Neigbours except parent
Assign Neigbours to SendList
Neigbour is NOT Parent
True
Get ENTRY TimeStamp
Add Delay to TimeStamp
Create new RECEIVE JOIN_ADV ENTRY
Add new Entry to TaskList
SendList is Processed
False

Figure A.3.4 Join message processing

If the message type of the entry is not advertisement the controller process checks the type whether it is join type or not. Join type messages are implemented as flow chart represented in Figure A.3.4. Firstly, entry type is considered, if the entry is an receive entry, sender node and message are extracted from the entry. It is checked whether the node processed the message before or not. Path and query are taken and node adds itself to the path variable in the message. Parent is set as sender of the entry and query is added to the query table of the node. Conditional processes compare receiving node's data with the data in the query. If receiving node's data is equal or bigger than the data in the query, node creates a new join request message, gets time stamp of the entry, adds delay to the time stamp, create a new entry with join request message intended to send parent node and adds this new entry to the task list. New join advertisement message is also created simultaneously. New receive entries with parsing the list filled by finding the neighbors of sender node are added to the task list with adding processing and sending delay to the entries. If the Node's data does not meet the condition stated in the query, new advertisement message is composed and new receiving entries are created and added to the task list by adding delays to the time stamp of the main entry.

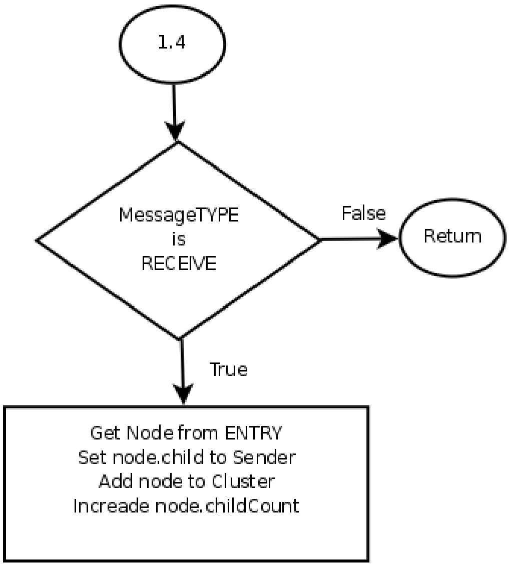

Figure A.3.5 Join request message processing

If the message type is join request and the entry type is receive, sender node is extracted, child node is assigned to the parent, node is added to the cluster and node's child count is increased (Figure A.3.5).

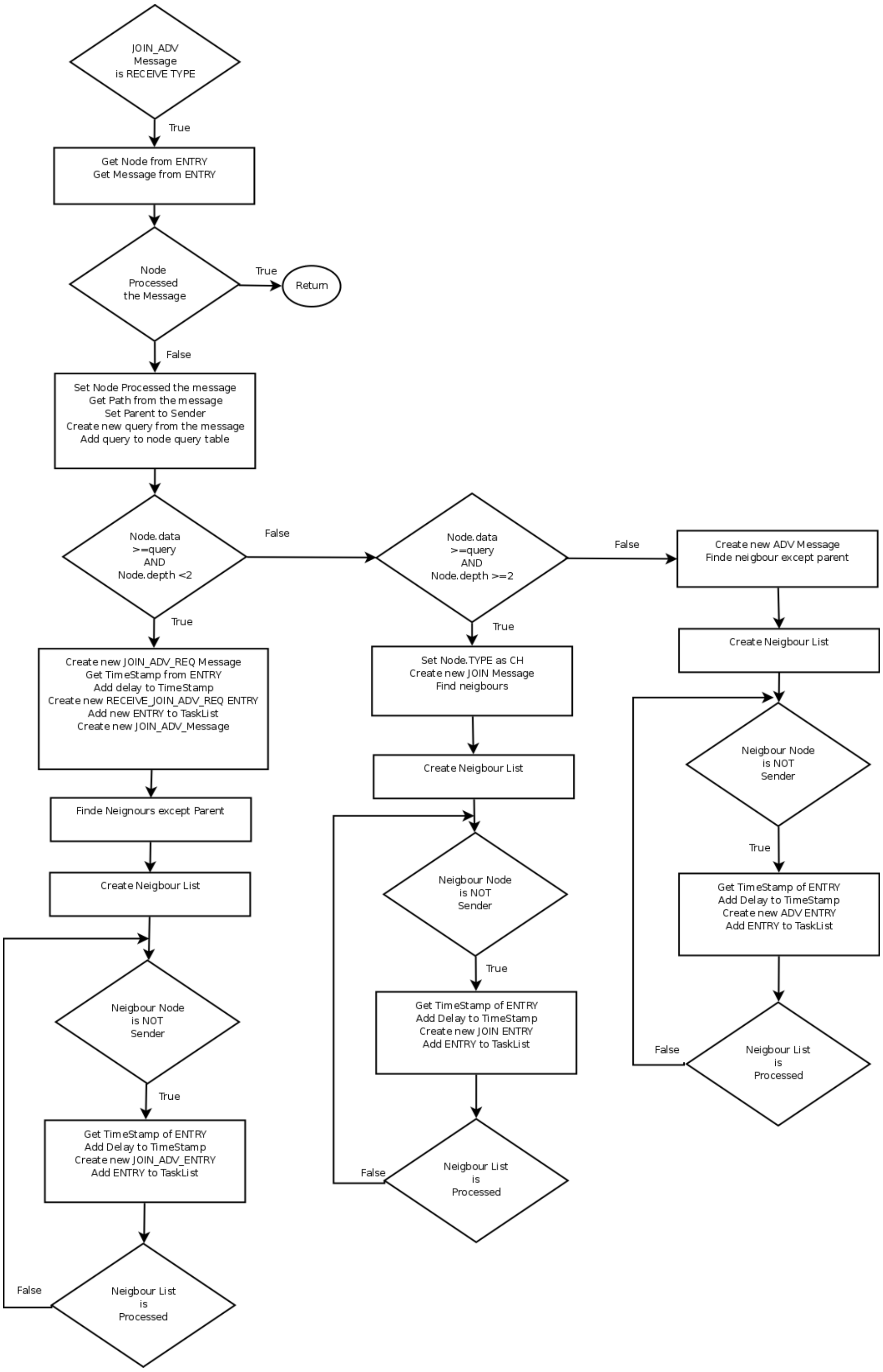

Figure A.3.6 Join advertisement message processing

If the message is in join advertisement message, entry type is controlled. Sender node and message is extracted from entry if node did not process the message before. If the node's data bigger than the query and the depth variable is less than 2, new join advertisement request message is created, delay is added to the time stamp of the main entry, a new receive type join advertisement request entry is created and this entry is added to the task list. Join advertisement request entry is sent to the parent node and a new join advertisement entry is created for propagating the query in the cluster. In addition, receive entries are created considering the neighboring nodes of the sender node and those entries are added to the task list by adding delay to the time stamp of the entry.

In the condition of node's data is bigger than the query and node's depth variable is bigger than 2. Node is set as cluster head, new join message is created, and receive entries considering the node's neighbors are inserted in to the task list.

In the condition of node does not satisfy the query, node creates a new advertisement message and adds itself to the path variable in the message. Receive entries considering the neighboring nodes are created with added time delays and those entries are added to the task list (Figure A.3.6).

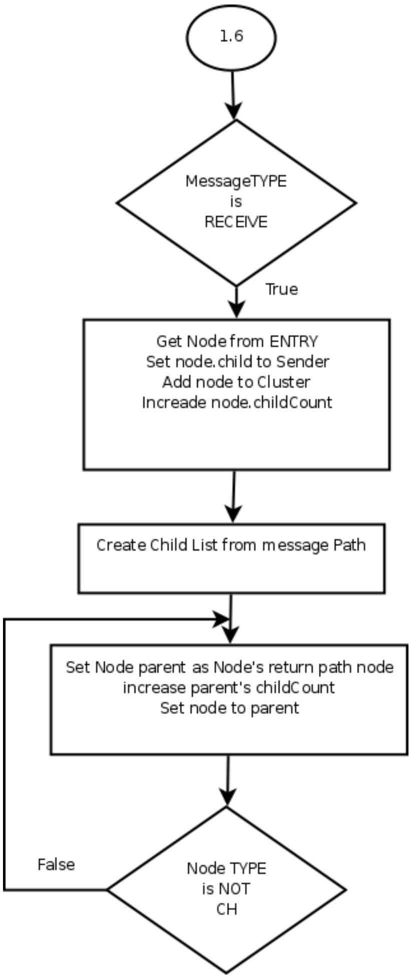

Figure A.3.7 Join advertisement request message processing

Join advertisement request messages are processed as in Figure 4.2.2.7. If the message type is join advertisement message and the entry type is receive, sender node is extracted from the entry, node is set as child node for the node that created the entry, node is added to the cluster and node child count is increased by 1. Until the node type is cluster head, nodes parent node is set as return path node, parent's child count is increased by 1, and node is set as parent.

A.4 Selecting new cluster head

In the propose model 1, new cluster head selection algorithm is developed by considering the locations of the nodes. The node which is closer to the average of X and Y values of the nodes in the cluster is assigned as new cluster head.

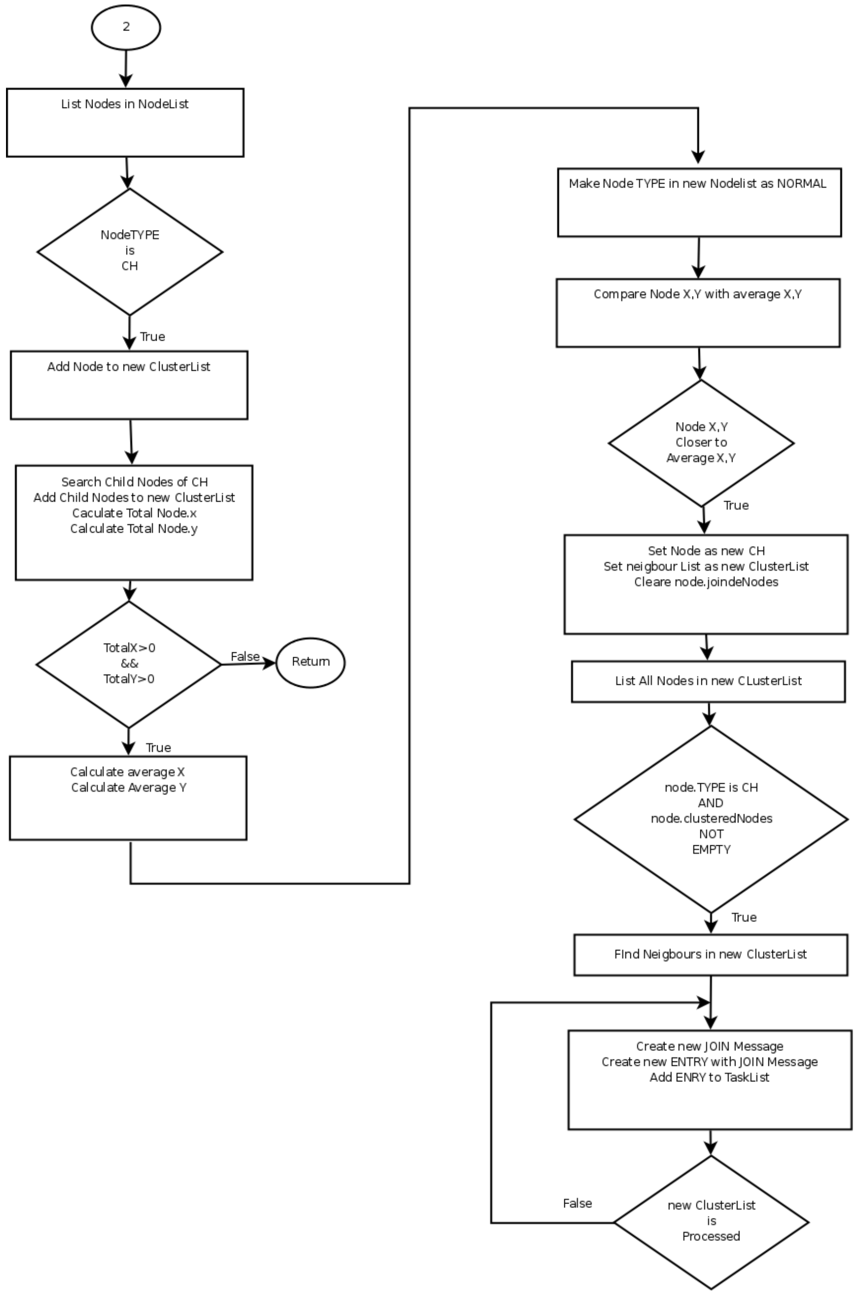
2
List Nodes in NodeList
NodeTYPE is CH
True
Add Node to new ClusterList
Search Child Nodes of CH
Add Child Nodes to new ClusterList
Caculate Total Node.x
Calculate Total Node.y
TotalX>0 && TotalY>0
False
Return
True
Calculate average X
Calculate Average Y
Make Node TYPE in new Nodelist as NORMAL
Compare Node X,Y with average X,Y
Node X,Y Closer to Average X,Y
True
Set Node as new CH
Set neigbour List as new ClusterList
Cleare node.joindeNodes
List All Nodes in new CLusterList
node.TYPE is CH AND node.clusteredNodes NOT EMPTY
True
FInd Neigbours in new ClusterList
Create new JOIN Message
Create new ENTRY with JOIN Message
Add ENRY to TaskList
False
new ClusterList is Processed

Figure A.4.1 Selecting new cluster head

In the re-clustering phase, all nodes in the simulation are tracked until a node with cluster head type is found. The node is added in a temporary list and child of the cluster head nodes are found. At the end of finding nodes in the cluster, all nodes total X and Y values are calculated. Calculated values are divided by the cluster node count and the average X and Y values are found. Node types of the nodes in the new temporary node list are set as normal type. Nodes X and Y values are compared with the average X and Y values. The node that has the shortest Euclidean distance to the average X and Y values is selected as new cluster head. That node's type is changed to cluster head and its joined node table is deleted. A new advertisement message is created and this message is added to a send entry. Entry is added to the time line and the controlling process creates new receive type entries containing the nodes that are in range of the new cluster head (Figure A.4.1).

A.5 Generating Schedules

Scheduling is an iterative process started by finding a cluster head node in the nodes in simulation.

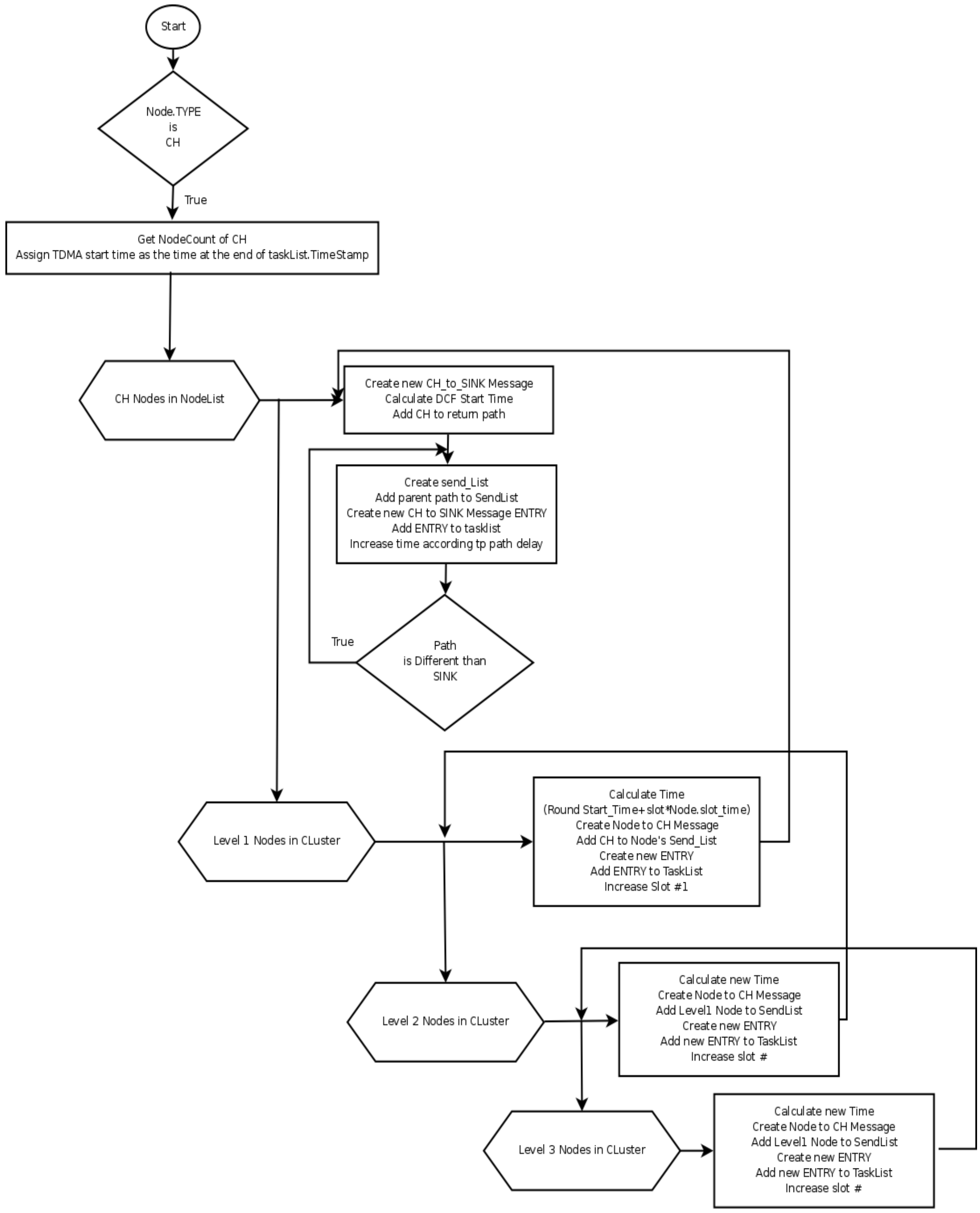
Start
Node.TYPE is CH
True
Get NodeCount of CH
Assign TDMA start time as the time at the end of taskList.TimeStamp
CH Nodes in NodeList
Create new CH_to_SINK Message
Calculate DCF Start Time
Add CH to return path
Create send_List
Add parent path to SendList
Create new CH to SINK Message ENTRY
Add ENTRY to tasklist
Increase time according tp path delay
True
Path is Different than SINK
Level 1 Nodes in CLuster
Calculate Time
(Round Start_Time+slot*Node.slot_time)
Create Node to CH Message
Add CH to Node's Send_List
Create new ENTRY
Add ENTRY to TaskList
Increase Slot #1
Level 2 Nodes in CLuster
Calculate new Time
Create Node to CH Message
Add Level1 Node to SendList
Create new ENTRY
Add new ENTRY to TaskList
Increase slot #
Level 3 Nodes in CLuster
Calculate new Time
Create Node to CH Message
Add Level1 Node to SendList
Create new ENTRY
Add new ENTRY to TaskList
Increase slot #

Figure A.5.1 Scheduling

In algorithm, cluster head nodes are found in the node list. Clusters node count is calculated by adding cluster head node's child nodes and their child nodes up to the third level of depth. Last task's time stamp is taken as the start time of the TDMA scheduling. Slot time is calculated as (round_time / cluster node_count +1). Total slots are distributed to the all child nodes in the cluster with three iterative loops. Each loop finds the child nodes and slots are assigned to the nodes calculating the child nodes of child nodes. Each parent node assigned slots according to their number of child nodes.

A.6 Cluster maintenance

After distributing the schedules to the nodes, cluster nodes start sending messages in their time slots. This messaging lasts until the cluster life time ends. Cluster life time is calculated as rountime x roundcount. Until cluster life time duration some management operations should be taken into account. The process starts by, processing the last entry produced after the scheduling phase.

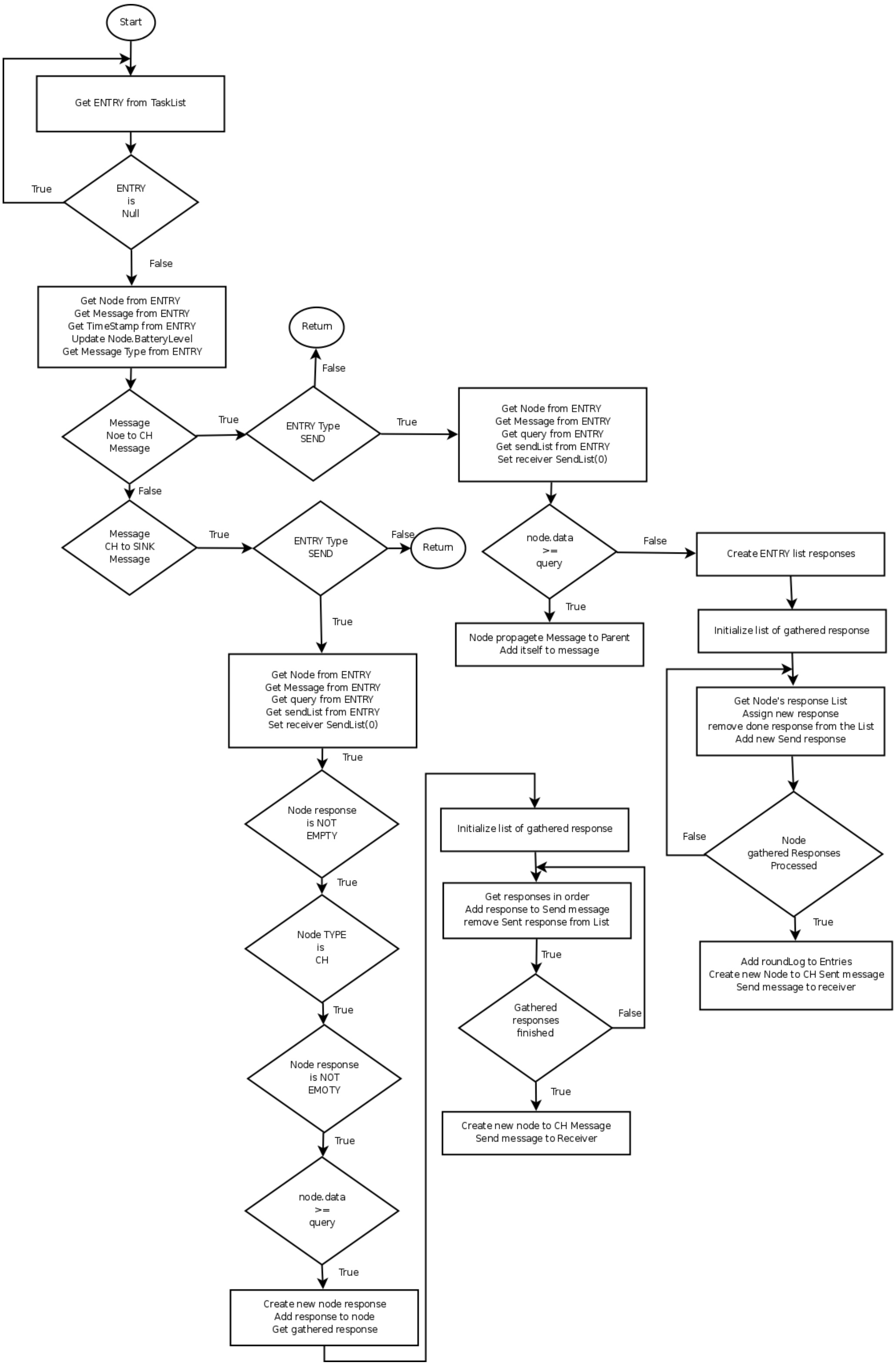

Figure A.6.1 Cluster maintenance

The last entry in the task list is treated. Sender node, message, time stamp and message type is extracted. If the message type is node to cluster head message and entry type is send, node, message, query, send list and receiver are extracted from the entry. If the data of the node is bigger than the query, Node creates a message that is to be sent to the parent node. Node creates an entry that has a time stamp which is equal to its time slot.

If the message type is cluster head to sink message and the entry type is send, node, message, query and send list extracted from the entry. In addition, receiver of the message is assigned as the first node in the send list. If node type is cluster head and node response is not empty and node's data is bigger than the query; new node response is created, this response added to the node and gathered responses of the node are extracted. A send message is composed with the responses and receive entries are created to send message to the sink node through the path variable. Receive entries are added to the task list with the expected delays added.