ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

ADDIS ABABA INSTITUTE OF TECHNOLOGY

ELECTRICAL & COMPUTER ENGINEERING DEPARTMENT

DESIGN AND PERFORMANCE EVALUATION OF POWER-AWARE ROUTING PROTOCOLS FOR WIRELESS SENSOR NETWORKS – GAICH AND GCH

By

Seifemichael Bekele Amsalu

October 2011

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

ADDIS ABABA INSTITUTE OF TECHNOLOGY

ELECTRICAL & COMPUTER ENGINEERING DEPARTMENT

Design and Performance Evaluation of Power-aware Routing Protocols for Wireless Sensor Networks – GAICH and GCH

By

Seifemichael Bekele Amsalu

Advisor

Dr. Ing. Dereje Hailemariam

A thesis submitted to the School of Graduate Studies of Addis Ababa
University in partial fulfillment of the requirements for the degree of
Masters of Science in Communication Engineering

October 2011

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

DESIGN AND PERFORMANCE EVALUATION OF POWER-AWARE ROUTING PROTOCOLS FOR WIRELESS SENSOR NETWORKS - GAICH AND GCH

By

Seifemichael Bekele Amsalu

ADDIS ABABA INSTITUTE OF TECHNOLOGY

APPROVAL BY BOARD OF EXAMINERS

Dr. Ing. Getahun Mekuria	___________________
Chairman, Dept. of Graduate Committee	Signature
Dr. Ing. Dereje Hailemariam	___________________
Advisor	Signature
___________________	___________________
Internal Examiner	Signature
___________________	___________________
External Examiner	Signature

Declaration

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been fully acknowledged.

Seifemichael Bekele Amsalu

Name

Signature

Place Addis Ababa, Ethiopia

Date of submission

This thesis has been submitted for examination with my approval as a university advisor.

Dr. Ing. Dereje Hailemariam

Advisor's name

Signature

Acknowledgement

Foremost, I would like to express my deepest gratitude to my thesis advisor *Dr. Ing. Dereje Hailemariam* for his continuous guidance, suggestion, advice, patience, critique and encouragement as well as all the knowledge he gave me during the course of this work.

Besides my advisor, I sincerely thank all the professors who have taught me, for the lesson that has been delivered. My special gratitude goes to *Dr. Ing. Hailu Ayele* for all the invaluable advises that he gave me during my stay in this program.

I would also like to take this opportunity to thank my colleagues for their help and encouragement to complete this thesis work. Every bit of it was useful and appreciated. Especial thanks goes to my friends *Anteneh Muhye*, *Fitsum Assamnew*, *Getachew Teshome* and *Wondimu Kassahun*.

Last but not the least, I would like to thank my family: my mother, father and sister for their valuable support and encouragement throughout my life.

Abstract

In recent years, the advancements in wireless communications and electronics have enabled the development of low-cost, low-power and multifunctional wireless sensor networks (WSNs). As nodes in sensor networks are equipped with a limited power source, efficient utilization of power is a very important issue in order to extend the network lifetime. It is for these reasons that researchers are currently focusing on the design of power-aware protocols and algorithms for sensor networks.

In this thesis, two routing protocols that provide efficient energy management for WSNs are proposed. The first protocol, GAICH (Genetic Algorithm Inspired Clustering Hierarchy), makes use of genetic algorithm to create optimum clusters in terms of energy consumption. The other one, GCH (Grid Clustering Hierarchy), creates clusters by forming virtual girds, where nodes share the role of cluster head in a round-robin fashion. These protocols have been implemented in MATLAB using a standard radio energy dissipation model that is used for the simulation of WSNs. Performance comparison has been made with two of the existing routing protocols: LEACH and Direct Transmission, on different performance metrics. Simulation results show that GAICH and GCH are better than LEACH in the total packets sent to the base station and network lifetime. Moreover, different techniques for optimizing energy consumption in WSNs are suggested.

Keywords: Wireless sensor networks, Energy efficiency, Clustering hierarchy, Genetic algorithm, Grid, Routing protocol

Table of Contents

Declaration I

Acknowledgement II

Abstract III

Table of Contents IV

List of Figures VI

List of Tables VIII

Acronyms IX

1. Introduction 1

- 1.1. Problem Statement 2
  - 1.1.1. General Objective 3
  - 1.1.2. Specific Objectives 3
- 1.2. Scope and Limitation 4
- 1.3. Organization 4

2. Literature Review 5

- 2.1. WSNs Communication Architecture and Design Factors 5
  - 2.1.1. WSNs Communications Architecture 5
  - 2.1.2. General Design Factors and Requirements 6
- 2.2. WSNs Protocol Stack 12
- 2.3. Energy Saving Routing Protocols 13
  - 2.3.1. Data-centric protocols 14
  - 2.3.2. Location-based protocols 21
  - 2.3.3. Hierarchical protocols 22
- 2.4. Low-Energy Adaptive Clustering Hierarchy 24

3. Genetic Algorithm 33

3.1. Operators of Genetic Algorithm 34

3.2. Basic Genetic Algorithm 39

4. Design and Implementation 41

4.1. Genetic Algorithm Inspired Clustering Hierarchy 42

4.2. Grid Clustering Hierarchy 51

5. Simulation and Results 56

5.1. Simulation Models 56

5.1.1. Channel Propagation Model 56

5.1.2. Radio Energy Dissipation Model 59

5.1.3. Beamforming Energy Model 62

5.2. Simulation Set-up 63

5.3. Simulation Results 65

5.3.1. Fixed Base Station Location 66

5.3.2. Varying the Base Station Location 72

6. Conclusions and Recommendations 74

6.1. Conclusions 74

6.2. Recommendations 75

References 77

List of Figures

Figure 2.1: Sensor nodes scattered in a sensor field 6

Figure 2-2: The components of a sensor node 8

Figure 2-3: The sensor networks protocol stack 13

Figure 2-4: The implosion problem. 16

Figure 2-5: The overlap problem. 17

Figure 2-6: SPIN Protocol. 18

Figure 2-7: Directed diffusion protocol phases 20

Figure 2-8: Chaining in PEGASIS 24

Figure 2-9: The LEACH protocol for microsensor networks 25

Figure 2-10: Dynamic clusters: a) Cluster head nodes= $\mathbf{C}$ at time $\mathbf{t1}$ (b) Cluster head nodes = $\mathbf{C}$' at time $\mathbf{t1 + d}$ 26

Figure 2-11: Time line showing LEACH operation 27

Figure 2-12: Flow-graph of the distributed cluster formation algorithm for LEACH 29

Figure 2-13: Flow-graph of the steady-state operation for LEACH 31

Figure 2-14: Radio interference. Node A's transmission to node B corrupts any transmission to node C 32

Figure 3-1: A single-point crossover 38

Figure 3-2: An example of Mutation 38

Figure 4-1: Flowchart of the Genetic Algorithm that runs on the Base Station. The outputs are the Cluster heads and the number of rounds $\boldsymbol{RCi}$ for the current configuration. 50

Figure 4-2: Flowchart of the operations of GAICH. Here the cluster formation and data transmission parts are the same as LEACH. 51

Figure 4-3: Three level virtual gridding in GCH for a 100-node random test network where the base station is located at the center of the network, at location (x=50, y=50). (a) First level gridding. (b) Second level gridding (c) Third level gridding 54

Figure 4-4: Flowchart of the operations of GCH. 55

Figure 5-1: First order radio model (Radio energy dissipation model) 59

Figure 5-2: Block diagram of the beamforming algorithm. The individual sensor signals, $Si[n]$ are filtered with weighting filters, $Wi[n]$ to get the beamformed signal, yn 63

Figure 5-3: 100-node random test network. The base station is located at the center of the network, at location (x=50, y=50). 64

Figure 5-4: Number of packets sent to base station per round 67

Figure 5-5: The total packets received at the base station over number of rounds. 68

Figure 5-6: The total packets sent to the base station without aggregation over number of rounds. 69

Figure 5-7: Number of dead nodes per round. 70

Figure 5-8: Number of nodes alive per total packets sent to the base station. 71

Figure 5-9: Total amount of energy dissipated in the system. 72

List of Tables

Table 4-1: Chromosome representation with two dimensional arrays with the first row representing the role of the sensor nodes and the second containing the corresponding node IDs. 43

Table 5-1: Radio characteristics and parameter values 64

Table 5-2: Characteristics of the test network. 65

Table 5-3: Lifetimes of the different protocols where the base station is at the center of the network 69

Table 5-4: Performance of the protocols as the base station location is varied. 73

Acronyms

ACQUIRE	Active query forwarding in sensor networks
ADC	Analog-to-digital converter
APTEEN	Adaptive threshold sensitive energy efficient sensor network protocol
BS	Base station
CADR	Constrained anisotropic diffusion routing
CDMA	Code division multiple access
CSMA	Carrier sense multiple access
EA	Evolutionary algorithm
GA	Genetic algorithm
GAF	Geographic adaptive fidelity
GAICH	Genetic algorithm inspired clustering hierarchy
GCH	Grid clustering hierarchy
GEAR	Geographic and energy aware routing
GPS	Global positioning system
GPSR	Greedy perimeter stateless routing
IP	Internet protocol
LEACH	Low-energy adaptive clustering hierarchy
LMS	Least mean squared
MAC	Medium access control

MECN	Minimum energy communication network
MEMS	Micro electro-mechanical systems
MSE	Mean squared error
NEMS	Nanoscale electro-mechanical systems
PEGASIS	Power-efficient gathering in sensor information systems
QoS	Quality of service
SA-1100	StrongARM-1100 processor
SCHC	Sum of the cluster heads centrality
SCHD	Sum of cluster heads density
SCHDBS	Sum of cluster head distances to base station
SCHRE	Sum of the cluster heads residual energy
SMECN	Small minimum energy communication network
SNR	Signal-to-noise ratio
SPIN	Sensor protocols for information via negotiation
TDMA	Time division multiple access
TEEN	Threshold sensitive energy efficient sensor network protocol
WINS	Wireless integrated network sensors
WSN	Wireless sensor networks
µAMPS	Micro-adaptive multi-domain power-aware sensors

Introduction

Wireless communication makes possible new network topologies that would, otherwise, be restricted for deployment in wired infrastructures. This possibility creates many challenges for wireless network designers. One of the challenges they face is the highly flexible, time varying nature of the network. This increases the demand for more adaptive network protocol stacks. In developing protocol stacks for wireless networks, designers must focus not only on dynamic parameters, which may change in real-time and require instantaneous attention, but also on characteristics such as network lifetime, latency, mobility and required signal bandwidth of that particular class of wireless network.

Recent advancements in wireless communications and electronics have enabled the development of low-cost, low-power, multi-functional sensor nodes that are small in size and able to communicate in short ranges. These tiny sensor nodes, which consist of sensing, data processing, and communicating components, are the basis of sensor networks. Sensor networks represent a significant improvement over traditional sensors through a number of high-density technologies, including micro electromechanical systems (MEMS) and nanoscale electromechanical systems (NEMS) [1] [2] [3].

A sensor network is composed of a large number of sensor nodes that are densely deployed either inside a phenomenon or very close to it. The position of sensor nodes need not be engineered or predetermined. This allows random deployment in inaccessible terrains or disaster relief operations. However, this also means that sensor network protocols and algorithms must possess self-organizing capabilities [4]. Another unique feature of sensor networks is the cooperative nature of sensor nodes. Sensor nodes are fitted with an onboard processor. Instead of sending the raw data to the nodes responsible for the fusion, they use their processing abilities to locally carry out simple computations and transmit only the required or partially processed data.

These features of sensor networks make possible a wide range of applications. Some of the application areas are health, military, and home. For example, in military, the rapid deployment, self-organization, and fault tolerance characteristics of sensor networks make them very promising sensing techniques for military command, control, communications, computing, intelligence, surveillance, reconnaissance, and targeting systems[5] [6]. In health, sensor nodes can be deployed to monitor patients and assist disabled patients [7]. Some other applications may include monitoring product quality, managing inventory, or monitoring disaster areas [8] [9] [10].

1.1. Problem Statement

Realization of sensor network applications requires wireless ad-hoc networking techniques. Although many protocols and algorithms have been proposed for traditional wireless ad hoc networks, they are not well suited to the unique features and application requirements of sensor networks. For illustration, the differences between sensor and ad-hoc networks are presented below [11]:

- The number of sensor nodes in a sensor network can be higher than that in an ad-hoc network.
- Nodes in a sensor network are densely deployed.
- Sensor nodes are more prone to failures.
- The topology of a sensor network changes very frequently.
- Nodes in a sensor network are limited in power, computational capability, and memory.
- Sensor nodes may not have global identification because of the large amount of overhead and large number of sensors.

The wireless sensor node, being a microelectronic device, can only be equipped with a limited power source. In most application scenarios, replenishment of power supplies is challenging, if not impossible. Therefore, sensor node lifetime shows a strong dependence on battery lifetime.

In multi-hop wireless sensor networks (WSNs), each node plays the dual role of data generation and data routing. The malfunctioning of a few nodes can cause significant topological changes requiring rerouting of packets and reorganization of the network. Since these operations increase power consumption, power management techniques are very essential here.

Recently, researchers have focused on the design of power-aware protocols and algorithms for sensor networks to address the problems stated above. The role of a sensor node in a sensor field is to detect events, perform quick local data processing, and then transmit the data. Power consumption can hence be attributed to three domains: sensing, communication, and data processing [2].

1.1.1. General Objective

The objective of this thesis is to devise and simulate routing protocols that provide efficient energy management mechanisms for WSNs. The proposed protocols will be simulated and compared with two existing protocols: LEACH and Direct Transmission, based on different performance metrics. Afterwards, different techniques will be suggested to optimize energy consumption in WSNs.

1.1.2. Specific Objectives

In this thesis, two energy efficient routing protocols are proposed for WSNs. The protocols are based on genetic algorithm and grid clustering hierarchy. The protocols, GAICH and GCH, will be implemented in MATLAB based on a standard radio energy dissipation model that is used for the simulation of WSNs. The following are the specific objectives of this work:

- Formulating and designing the protocols
- Implementing the protocols in MATLAB and running tests with simulation.
- Evaluating performances of the protocols based on the metrics: network lifetime, total packets received at the base station and amount of energy consumed per successful data report.
- Comparing their performance with a standard clustering protocol, LEACH.
- Investigating different methods for optimizing energy consumption in WSNs.

1.2. Scope and Limitation

In this thesis, performance evaluation of the proposed protocols is made based on three metrics: network lifetime, total number of packets received at the base station and amount of energy consumed per successful data report. As the work considers only the network layer from the protocol stack, the underlying MAC protocol is abstracted by the radio dissipation model. Consequently, other subjects such as delay of delivering the data packets, overhead of running the algorithms and data loss due to channel behaviors are not taken into consideration.

The optimization of energy dissipation is done using genetic algorithm in GAICH. This results in local optimum as the number of generations and population size taken are limited to reduce the computation cost in the base station. Furthermore, the mobility of sensor nodes isn’t considered.

1.3. Organization

The thesis is organized as follows. The second Chapter contains the literatures surveyed on WSNs. WSN communication architecture and design factors, protocol stack, energy efficient routing protocols and Low-Energy Adaptive Clustering Hierarchy (LEACH) are discussed in this chapter.

Since one of the proposed routing protocols is based on genetic algorithm, discussion on Genetic Algorithm (GA) has been made in Chapter 3.

Chapter 4 is where the design and implementation of the two power aware hierarchical routing protocols, Genetic Algorithm Inspired Clustering Hierarchy (GAICH) and Grid Clustering Hierarchy (GCH), is presented.

In Chapter 5, tests carried out and results obtained for the two routing protocols are presented. Comparison with existing protocols, LEACH and Direct Transmission, are made based on different performance metrics.

Finally, conclusions and recommendations for future work on energy efficient routing protocols are presented in Chapter 6.

Literature Review

In this chapter, the literature survey that is done to support the work on devising energy efficient routing protocols is discussed thoroughly. The overall communication architecture and protocol stack of the WSN are explained in the first sections. Despite energy saving can be achieved at different layers of the WSN protocol stack, the focus of this thesis is limited to energy saving routing protocols. So, in the subsequent sections, energy efficient routing protocols are described. As the work is based on hierarchical clustering, LEACH which is a cluster based hierarchical routing protocol for WSN is also discussed in detail.

2.1. WSNs Communication Architecture and Design Factors

In this section, a general network topology of WSN is explained in detail. When we design a sensor network, there are several issues that should be taken into account. Here, the important design factors and requirement are discussed.

2.1.1. WSNs Communications Architecture

As it is mentioned in Chapter 1, a WSN is a network made of a numerous number of sensor nodes with sensing, communication and data processing capabilities. These sensor nodes are usually scattered in a sensor field situated far from the user as shown in Figure 2-1. Each of these scattered sensor nodes has the capabilities to collect data and route data back to the sink. Data are routed back to the sink by a multi-hop infrastructure-less architecture.

The sink may communicate with the task manager node via Internet or satellite. The design of the sensor network as described by Figure 2-1 is influenced by many factors, including fault tolerance, scalability, production costs, operating environment, sensor network topology, hardware constraints, transmission media, and power consumption.

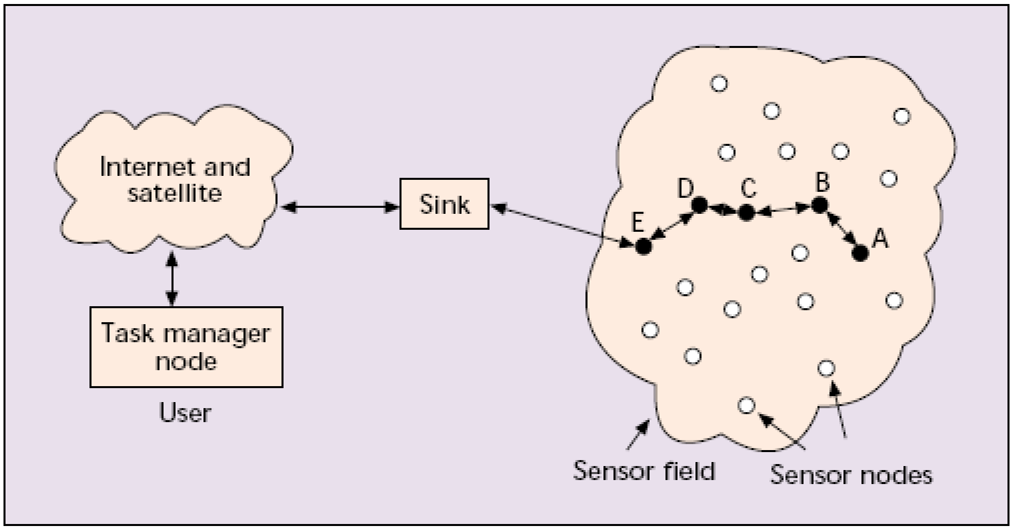


Figure 2.1: Sensor nodes scattered in a sensor field [2]

2.1.2. General Design Factors and Requirements

In this sub-section, the design factors of sensor networks and sensor nodes are explained in detail. Many researchers have addressed many designing factors which include fault tolerance/reliability, scalability, production costs, operating environment, sensor network topology, hardware constraints, transmission media, and power consumption. These factors are important because they serve as a guideline to design a protocol or an algorithm for sensor networks. In addition, these influencing factors can be used to compare different schemes.

In addition to the design factors, there are many requirements that should be addressed depending on application. The requirements may include data aggregation/data fusion, security, self-configuration, network dynamics, Quality of Service, Coverage and Connectivity. The design factors and requirement are discussed below.

Fault Tolerance / Reliability: - is the ability to maintain the overall functionality of the sensor network without any interruption due to sensor node failures [12] [13]. Sensor nodes may fail due to lack of power, physical damage or environmental interference. Reliability, $R_k(t)$, is modeled in [13] using the Poisson distribution to capture the probability of not having a failure within the time interval $(0, t)$:

$$R_k(t) = e^{-\lambda_k t} \tag{2-1}$$

Where λ_k is the failure rate of sensor node k and t is the time period

Scalability: - In the order of hundreds or thousands of sensor nodes may be deployed to studying a phenomenon of interest. Depending on the application, the number may reach extreme values. The algorithms that are used in sensor networks must be able to work with this number of nodes. They must also utilize the high density of the sensor networks. The density can range from few sensor nodes to few hundred sensor nodes in a region, which can be less than 10 m in diameter. The density μ can be calculated as in [14].

$$\mu(R) = (N \,.\, \pi R^2)/A \tag{2-2}$$

Where N is the number of scattered sensor nodes in region A, and R is the radio transmission range. Basically, $\mu(R)$ gives the number of nodes within the transmission radius of each node in region A.

Production Costs: - Since sensor networks consist of a large number of sensor nodes, the cost of a single node is very important to justify the overall cost of the network. If the cost of the network is more expensive than deploying traditional sensors, the sensor network is not cost-justified [15]. As a result, the cost of each sensor node has to be kept low.

Hardware Constraints: - A sensor node is made up of four basic components, as shown in Figure 2-2: a sensing unit, a processing unit, a transceiver unit, and a power unit. They may also have additional application-dependent components such as a location finding system, power generator, and mobilizer. Sensing units are usually composed of two subunits: sensors and analog-to-digital converters (ADCs). The analog signals produced by the sensors based on the observed phenomenon are converted to digital signals by the ADC, and then fed into the processing unit. The processing unit, which is generally associated with a small storage unit, manages the procedures that make the sensor node collaborate with the other nodes to carry out the assigned sensing tasks. A transceiver unit connects the node to the network.

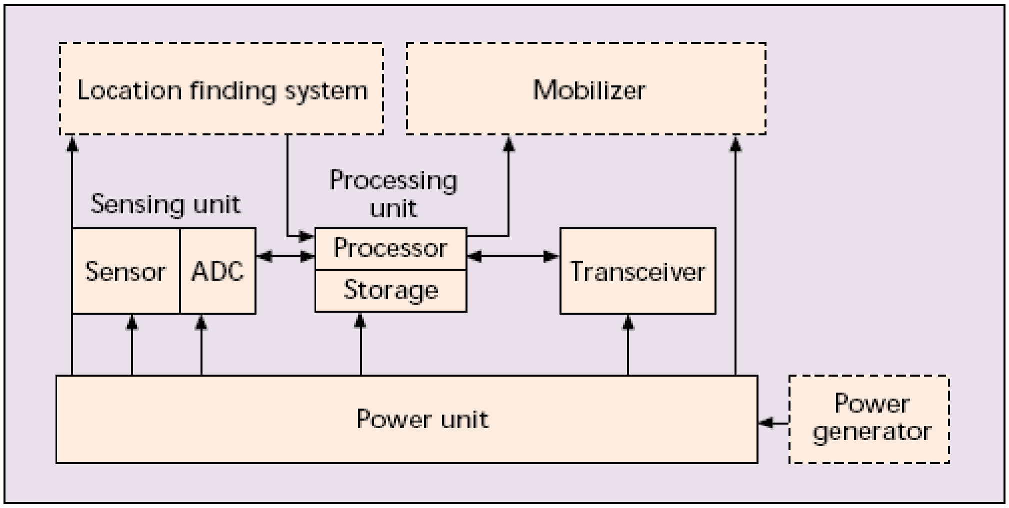


Figure 2-2: The components of a sensor node [2].

One of the most important components of a sensor node is the power unit. Power unit may be supported by power scavenging units such as solar cells. Most of the sensor network routing techniques and sensing tasks require knowledge of location with high accuracy. Thus, it is common that a sensor node has a location finding system. A mobilizer may sometimes be needed to move sensor nodes when it is required to carry out the assigned tasks. All of these subunits may need to fit into a matchbox-sized module. Hence, the size of sensor node is of a great design issue [16].

Apart from size, there are some other stringent constraints for sensor nodes. These nodes must consume extremely low power, operate in high volumetric densities, have low production cost, be dispensable and autonomous, operate unattended, and be adaptive to the environment [17].

Sensor Network Topology: - the topology of a network affects many of its characteristics like; latency, capacity, and robustness. Also, the complexity of data routing and processing depends on the network topology. Hundreds to several thousands of nodes are deployed throughout the sensor field. They are deployed within tens of centimeters of each other. The node densities may be as high as 20 nodes/m^2 [18].[18] Deploying a high number of nodes densely requires careful handling of topology maintenance. Issues related to topology maintenance and changes are examined in three phases [19].

- *Pre-deployment and deployment phase*: Sensor nodes can be deployed by dropping from a plane, delivered in an artillery shell, rocket, or missile, and placed one by one by either a human or a robot.

- *Post-deployment phase*: After deployment, topology changes due to change in sensor nodes' position, reachability (due to jamming, noise, moving obstacles, etc.), available energy, malfunctioning, and task details.

- *Redeployment of additional nodes phase*: Additional sensor nodes can be redeployed at any time to replace malfunctioning nodes or due to changes in task dynamics.

Environment: - Sensor nodes are densely deployed either very close to or directly inside the phenomenon to be observed. Therefore, they usually work unattended in remote geographic areas. They may be working in the interior of large machinery, at the bottom of an ocean, in a biologically or chemically contaminated field, in a battlefield beyond the enemy lines, and in a home or large building.

Transmission Media: **-** In a sensor network, communicating nodes are linked by a wireless medium. These links can be formed by radio, infrared, or optical media. To enable global operation of these networks, the chosen transmission medium must be available worldwide. Much of the current hardware for sensor nodes is based on RF circuit design. The *µAMPS* wireless sensor node described in [18] uses a Bluetooth-compatible 2.4 GHz transceiver with an integrated frequency synthesizer. The low-power sensor device described in [20] uses a single-channel RF transceiver operating at 916 MHz. The *Wireless Integrated Network Sensors (WINS)* architecture [16] also uses radio links for communication.

Another possible mode of internode communication in sensor networks is by infrared. Infrared communication is license-free and robust to interference from electrical devices. Infrared-based transceivers are cheaper and easier to build. Another interesting development is that of the *Smart Dust mote* [17] which is an autonomous sensing, computing, and communication system that uses the optical medium for transmission. Both infrared and optical require a line of sight between the sender and receiver.

Power Consumption: **-** the wireless sensor node can only be equipped with a limited power source (< 0.5 Ah, 1.2 V). The refilling of power resources might be impossible in some application scenarios. Therefore, a sensor node lifetime shows a strong dependence on battery lifetime. Hence, power management has basic importance. Consequently, researchers are currently focusing on the design of power-aware protocols and algorithms for sensor networks.

In other mobile and ad hoc networks, power consumption has been an important design factor, but not the primary consideration, simply because power resources can be replaced by the user. The emphasis is more on QoS provisioning than the power efficiency. In sensor networks though, power efficiency is an important performance metric, directly influencing the network lifetime. Application specific protocols can be designed by appropriately trading off other performance metrics such as delay and throughput with power efficiency.

Data Aggregation/Data Fusion: **-** is the task of reducing data size by summarizing the data into a set of meaningful information via computation while data are propagating through the WSN. As sensor networks are made of large number of sensor nodes, the network can easily be congested and flooded with information [21]. Hence; a solution to data congestion in sensor networks is to use computation to aggregate or fuse data within WSN, then transmit only the aggregated data to the controller. Many approaches within the context of WSNs are proposed to facilitate data aggregation/data fusion.

Security: **-** security aspects in WSNs have been focused on the centralized communications approaches. Some of the threats to a WSN are described in [22] [23] [24] and categorized as follows: Passive Information Gathering, False Node, Node Outage, Supervision of a Node, Node Malfunction, Message Corruption, Denial of Service, and Traffic Analysis. There is a need to develop distributed security approaches for WSN.

Self-Configuration: **-** it is essential for WSN to be self-organizing. Since the densely deployed sensor nodes in a sensor field may fail due to many reasons and nodes may join the network. On the other hand, sensor nodes work unattended in a dynamic environment; so they need to be self-configuring to establish a topology that supports communications under severe energy constraints. It is worth mentioning that self-configuration in WSN is an essential factor to maintain a WSN functions properly and serve its purpose [14] [25].

Quality of Service: - for some applications, data delivery within a bounded latency (i.e., time constrained applications) is of great importance; otherwise, the sensed data that delivered after certain latency will be useless. In other applications, the conservation of power is more important than the quality of the sent data. Hence; there is a tradeoff between the quality of service and the energy consumption depending on the applications [26] [27]

Coverage: - the sensor node's view of the environment that it is situated in is limited both in range and in accuracy. This means the ability of sensor nodes to cover physical area of the environment is limited [28] [29].

Connectivity: - a permanent connection between any two individual sensor nodes that are densely deployed in a sensor network defines the network connectivity. The connectivity is of great importance, since it influences communications protocols' design and data dissemination techniques. Also, it is worth mentioning that connectivity of sensor network may not prevent the network topology from being variable and the network size from reduction as a result of the death or failure of some sensor nodes due to the reasons mentioned earlier in the paper [28] [29].

Network dynamics: **-** in many applications, the movement of sensor nodes or sink is essential [30]. As a result, routing messages from or to moving nodes is more challenging since route stability becomes an important optimization factor, in addition to energy, bandwidth etc. Moreover, the specific sensed phenomenon may be either dynamic or stationary depending on the applications [31]. For example, in a target detection/tracking application, the event is dynamic whereas forest monitoring for early fire prevention is an example of static events. Monitoring static events allows the network to work in a reactive mode, simply generating traffic when reporting. Dynamic events in most applications require periodic reporting and consequently generate significant traffic to be routed to the sink.

2.2. WSNs Protocol Stack

The protocol stack implemented in the sensor nodes is shown in Figure 2-3. This protocol stack needs to combine power and routing awareness, integrate data with networking protocols, communicate power efficiently through the wireless medium, and promote cooperative efforts of sensor nodes. The protocol stack consists of the physical layer, data link layer, network layer, transport layer, application layer, power management plane, mobility management plane, and task management plane. The physical layer addresses the needs of simple but robust modulation, transmission, and receiving techniques. Since the environment is noisy and sensor nodes can be mobile, the medium access control (MAC) protocol must be power-aware and able to minimize collision with neighbors' broadcasts. The network layer takes care of routing the data supplied by the transport layer. The transport layer helps to maintain the flow of data if the sensor networks application requires it. Depending on the sensing tasks, different types of application software can be built and used on the application layer. In addition, the power, mobility, and task management planes monitor the power, movement, and task distribution among the sensor nodes. These planes help the sensor nodes coordinate the sensing task and lower overall power consumption.

The power management plane manages how a sensor node uses its power and manages its power consumption among the three operations: sensing, computation, and wireless communications.

The power management plane manages how a sensor node uses its power. For example, the sensor node may turn off its receiver after receiving a message from one of its neighbors. This is to avoid getting duplicated messages. Also, when the power level of the sensor node is low, the sensor node broadcasts to its neighbors that it is low in power and cannot participate in routing messages. The remaining power is reserved for sensing. The mobility management plane detects and registers the movement of sensor nodes, so a route back to the user is always maintained, and the sensor nodes can keep track of who their neighbor sensor nodes are. By knowing who the neighbor sensor nodes are, the sensor nodes can balance their power and task usage. The task management plane balances and schedules the sensing tasks given to a specific region. Not all sensor nodes in that region are required to perform the sensing task at the same time. As a result, some sensor nodes perform the task more than others depending on their power level. These management planes are needed so that sensor nodes can work together in a power efficient way, route data in a mobile sensor network, and share resources between sensor nodes.

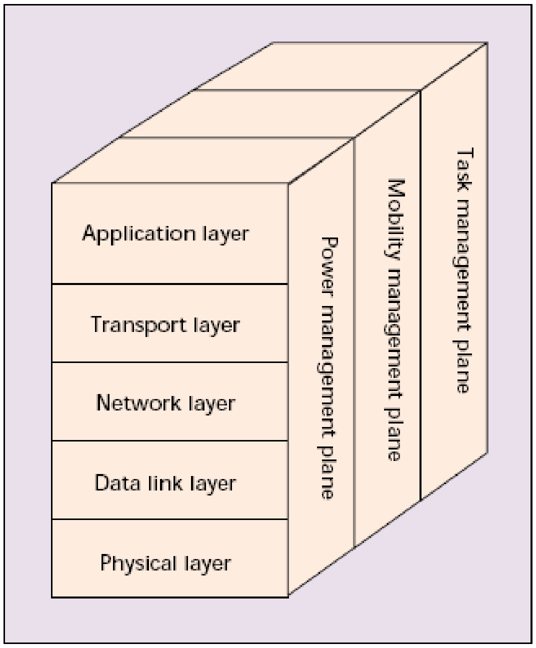


Figure 2-3: The sensor networks protocol stack [2].

Energy saving in WSN can be done at different levels of the protocol stack. We can save energy starting from the design of the senor node to the application software that runs on it. Since the work focuses on designing and implementing energy efficient routing protocols, the following section discusses thoroughly energy saving routing protocols that are used in different application areas.

2.3. Energy Saving Routing Protocols

Routing in sensor networks is very challenging due to several characteristics that distinguish them from contemporary communication and wireless ad-hoc networks. First of all, it is not possible to build a global addressing scheme for the deployment of sheer number of sensor nodes. Therefore, classical IP-based protocols cannot be applied to sensor networks. Second, in contrary to typical communication networks almost all applications of sensor networks require the flow of sensed data from multiple regions (sources) to a particular sink. Third, generated data traffic has significant redundancy in it since multiple sensors may generate same data within the vicinity of a phenomenon. Such redundancy needs to be exploited by the routing protocols to improve energy and bandwidth utilization. Fourth, sensor nodes are tightly constrained in terms of transmission power, on-board energy, processing capacity and storage and thus require careful resource management.

In this section, the different types of routing protocols that are proposed recently are highlighted. Almost all of the routing protocols can be classified in the following main categories [32]:

- Data-centric
- Location-based
- Hierarchical

In addition, there are few distinct ones based on network flow or QoS awareness [33] [34] [35] [36]. Some protocols from each category are described and discussed here.

Data-centric protocols are query-based and depend on the naming of desired data, which helps in eliminating many redundant transmissions. Location-based protocols utilize the position information to relay the data to the desired regions rather than the whole network. Hierarchical protocols aim at clustering the nodes so that cluster heads can do some aggregation and reduction of data in order to save energy.

2.3.1. Data-centric protocols

In many applications of sensor networks, it is not feasible to assign global identifiers to each node due to the large number of nodes deployed. Such lack of global identification along with random deployment of sensor nodes makes it hard to select a specific set of sensor nodes to be queried. Therefore, data is usually transmitted from every sensor node within the deployment region with redundancy. Since this is very inefficient in terms of energy consumption, routing protocols that will be able to select a set of sensor nodes and utilize data aggregation during the relaying of data have been considered. This consideration has led to data centric routing, which is different from traditional address-based routing where routes are created between addressable nodes managed in the network layer of the communication stack. In data-centric routing, the sink sends queries to certain regions and waits for data from the sensors located in the selected regions. Since data is being requested through queries, attribute based naming is necessary to specify the properties of data. SPIN [37] is the first data-centric protocol, which considers data negotiation between nodes in order to eliminate redundant data and save energy. Later, Directed Diffusion [19] has been developed and has become a breakthrough in data-centric routing. Then, many other protocols have been proposed either based on Directed Diffusion [38][39][40] or following a similar concept [41][42][43][44]. The following lists of protocols are within the data-centric category.

- Flooding and Gossiping [45]
- Sensor protocols for information via negotiation(SPIN) [37]
- Directed diffusion [19]
- Energy-aware routing: Shah et al [43]
- Rumor routing [38]
- Gradient-based routing: Schurgers et al. [39]
- Constrained anisotropic diffusion routing (CADR) [40]
- COUGAR [42]
- Active query forwarding in sensor networks (ACQUIRE) [44]

Since most of the data-centric protocols are based on Flooding and Gossiping, SPIN and Directed diffusion, these three protocols are explained in detail.

I. **Flooding and Gossiping** [45]

Flooding and gossiping are two old techniques to relay data in sensor networks without the need for any routing algorithms and topology maintenance. In flooding, each sensor receiving a data packet broadcasts it to all of its neighbors and this process continues until the packet arrives at the destination or the maximum number of hops for the packet is reached. On the other hand, gossiping is a slightly enhanced version of flooding where the receiving node sends the packet to a randomly selected neighbor, which picks another random neighbor to forward the packet to and so on. Although flooding is very easy to implement, it has several drawbacks, see Figure 2-4 and 2-5 redrawn from [37]. Such drawbacks include:

- ***Implosion:*** Implosion is a situation where duplicated messages are sent to the same node. For example, if sensor node A has N neighbor sensor nodes that are also the neighbors of sensor node B, sensor node B receives N copies of the message sent by sensor node A.
- ***Overlap:*** If two nodes share the same observing region, both of them may sense the same stimuli at the same time. As a result, neighbor nodes receive duplicated messages.
- ***Resource blindness:*** The flooding protocol does not take into account the available energy resources. An energy resource aware protocol must take into account the amount of energy available to them at all times.

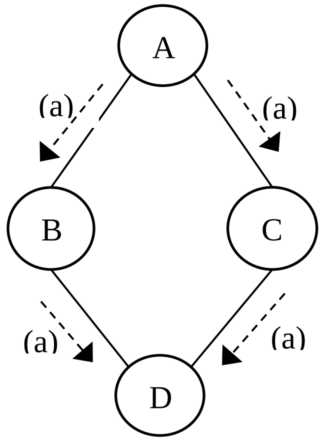


Figure 2-4: The implosion problem.

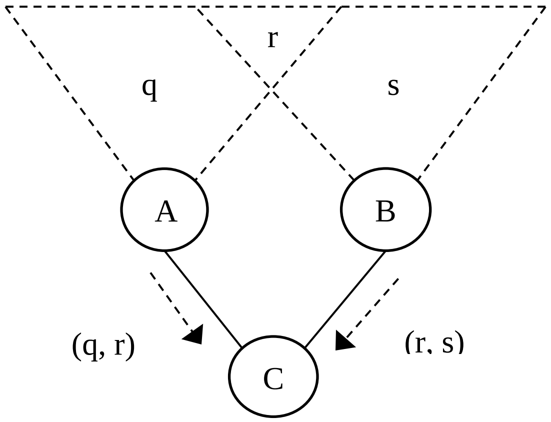


Figure 2-5: The overlap problem.

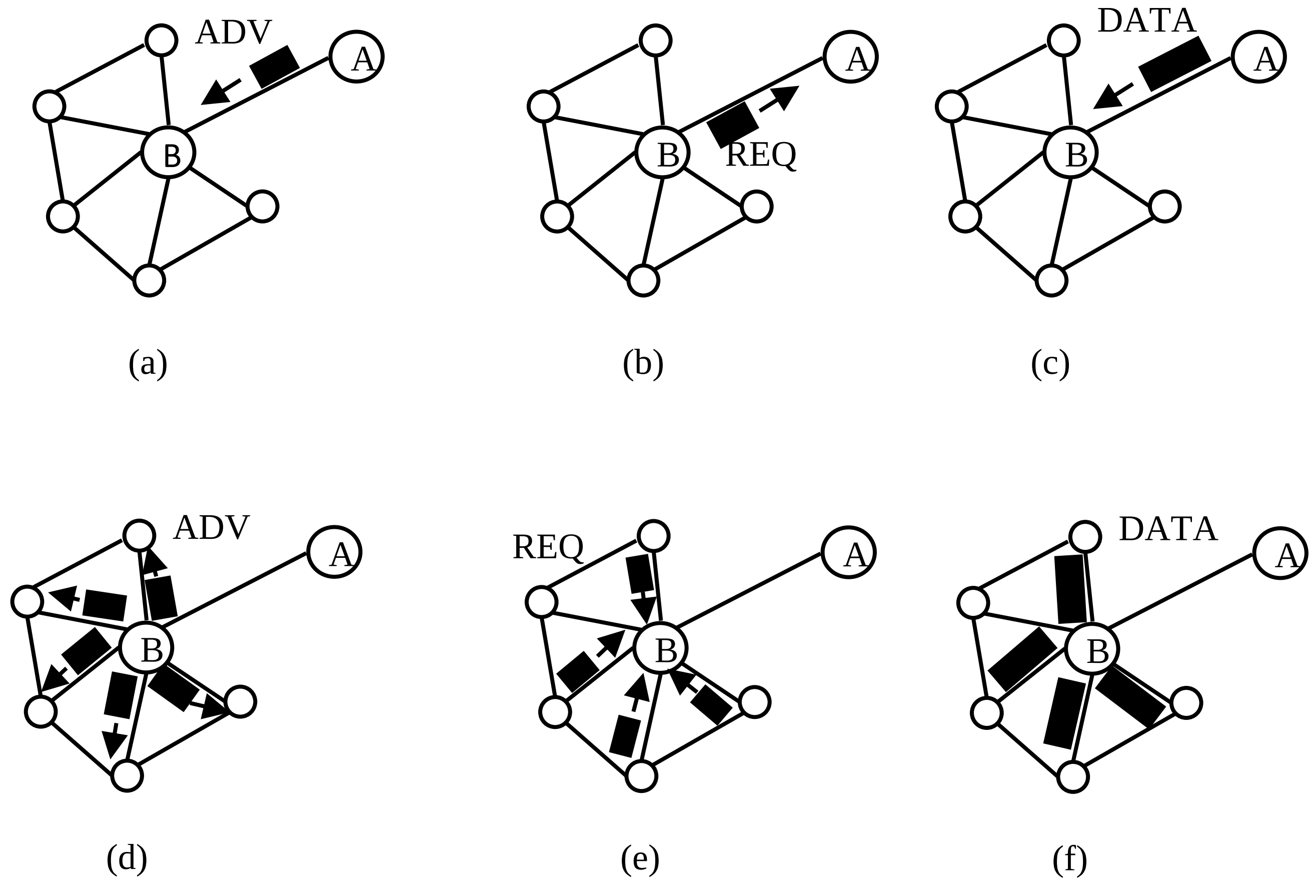
In Figure 2-4, we see the implosion problem, Node A starts by flooding its data to all of its neighbors. Then, D gets two copies of the same data eventually. The system wastes energy and bandwidth in unnecessary send and receive. The overlap problem is shown in Figure 2-5. Here, two sensors cover an overlapping geographic region. When these sensors flood their data to node C, C receives two copies of the data marked r and gets same copy of data from these sensors.

Gossiping avoids the problem of implosion by just selecting a random node to send the packet rather than broadcasting. However, this cause delays in propagation of data through the nodes.

II. Sensor Protocols for Information via Negotiation (SPIN) [37]

The SPIN family of protocols incorporates two key innovations that overcome the deficiencies in flooding; negotiation and resource adaptation. SPIN nodes negotiate with each other before transmitting data. Negotiation helps to ensure that only useful information will be transferred. To negotiate successfully, however, nodes must be able to describe or name the data they observe. These descriptors are referred as meta-data. Before transmission, meta-data are exchanged among sensors via a data advertisement mechanism. Each node upon receiving new data, advertises it to its neighbors and interested neighbors i.e., those who do not have the data, retrieve the data by sending a request message. SPIN's meta-data negotiation solves the problems of flooding such as redundant information passing, overlapping of sensing areas and resource blindness thus, achieving a lot of energy efficiency.

There are three messages defined in SPIN to exchange data between nodes. These are: ADV message to allow a sensor to advertise a particular meta-data, REQ message to request the specific data and DATA message that carry the actual data. Figure 2-6, redrawn from [37], summarizes the steps of the SPIN protocol. Here, Node A starts by advertising its data to node B (a). Node B responds by sending a request to node A (b). After receiving the requested data (c), node B then sends out advertisements to its neighbors (d), who in turn send requests back to B (e).Then, node B sends the requested data (f).



(a) (b) (c)

(d) (e) (f)

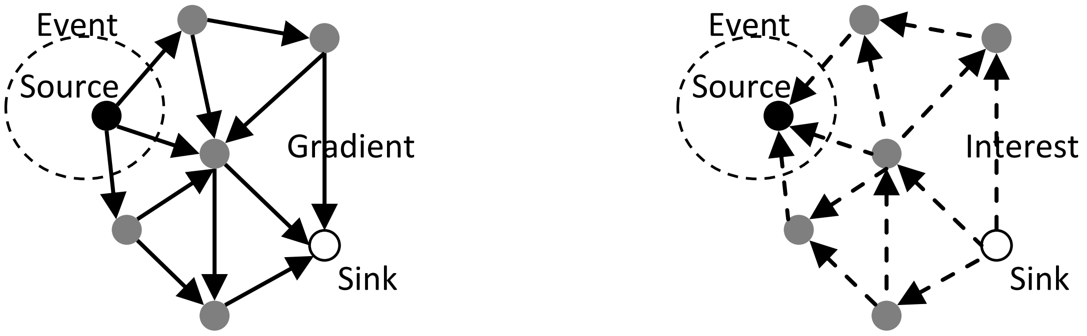
Figure 2-6: SPIN Protocol.

One of the advantages of SPIN is that topological changes are localized since each node needs to know only its single-hop neighbors. SPIN gives a factor of 3.5 less than flooding in terms of energy dissipation and meta-data negotiation almost halves the redundant data [37]. However, SPIN’s data advertisement mechanism cannot guarantee the delivery of data. For instance, if the nodes that are interested in the data are far away from the source node and the nodes between source and destination are not interested in that data, such data will not be delivered to the destination at all. Therefore, SPIN is not a good choice for applications such as intrusion detection, which require reliable delivery of data packets over regular intervals.

III. Directed Diffusion [19]

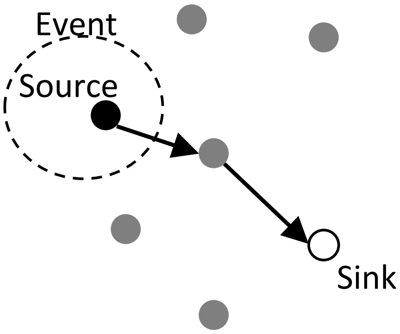
Direct Diffusion suggests the use of attribute-value pairs for the data and queries the sensors in an on demand basis by using those pairs. In order to create a query, an *interest* is defined using a list of attribute-value pairs such as name of objects, interval, duration, geographical area, etc. The interest is broadcast by a sink through its neighbors. Each node receiving the interest can do caching for later use. The interests in the caches are then used to compare the received data with the values in the interests. The interest entry also contains several gradient fields. A *gradient* is a reply link to a neighbor from which the interest was received. It is characterized by the data rate, duration and expiration time derived from the received interest’s fields. Hence, by utilizing interest and gradients, paths are established between sink and sources. Several paths can be established so that one of them is selected by reinforcement. The sink resends the original interest message through the selected path with a smaller interval hence reinforces the source node on that path to send data more frequently. Figure 2-7, redrawn from [19], summarizes the Directed Diffusion protocol.

Path repairs are also possible in Directed Diffusion. When a path between a source and the sink fails, a new or alternative path should be identified. For this, Directed Diffusion basically reinitiates reinforcement by searching among other paths, which are sending data in lower rates. Ganesan et al. [46] suggest employing multiple paths in advance so that in case of a failure of a path, one of the alternative paths is chosen without any cost for searching for another one. There is of course extra overhead of keeping these alternative paths alive by using low data rate, which will definitely use extra energy but more energy can be saved when a path fails and a new path should be chosen.



a) Interest propagation

b) Initial gradients setup



c) Data delivery along reinforced

Figure 2-7: Directed diffusion protocol phases

Directed Diffusion differs from SPIN in terms of the on demand data querying mechanism it has. In Directed Diffusion the sink queries the sensor nodes if a specific data is available by flooding some tasks. In SPIN, sensors advertise the availability of data allowing interested nodes to query that data. Directed Diffusion has many advantages. Since it is data centric, all communication is neighbor-to-neighbor with no need for a node addressing mechanism. Each node can do aggregation and caching, in addition to sensing. Caching is a big advantage in terms of energy efficiency and delay. In addition, Direct Diffusion is highly energy efficient since it is on demand and there is no need for maintaining global network topology.

However, directed diffusion cannot be applied to all sensor network applications since it is based on a query-driven data delivery model. The applications that require continuous data delivery to the sink will not work efficiently with a query-driven on demand data model. Therefore, Directed Diffusion is not a good choice as a routing protocol for the applications such as environmental monitoring. In addition, the naming schemes used in Directed Diffusion are application dependent and each time should be defined a priori. Moreover, the matching process for data and queries might require some extra overhead at the sensors.

2.3.2. Location-based protocols

Routing protocols for sensor networks may require location information for sensor nodes. In most cases location information is needed in order to calculate the distance between two particular nodes so that energy consumption can be estimated. Since there is no addressing scheme for sensor networks like IP-addresses and they are spatially deployed on a region, location information can be utilized in routing data in an energy efficient way. For instance, if the region to be sensed is known, using the location of sensors, the query can be diffused only to that particular region which will eliminate the number of transmission significantly. The following protocols are designed primarily for mobile ad hoc networks. However, they are also well applicable to sensor networks where there is less or no mobility.

- Minimum Energy Communication Network (MECN) and SMECN [47] [48]
- Geographic Adaptive Fidelity (GAF) [49]

- Geographic and Energy Aware Routing (GEAR), Yu et al. [50]
- Greedy Perimeter Stateless Routing (GPSR) [51]

2.3.3. Hierarchical protocols

Similar to other communication networks, scalability is one of the major design attributes of sensor networks. A single-gateway network can cause the gateway to overload with an increase in sensors density. Such overload might cause latency in communication and inadequate tracking of events. In addition, the single-gateway architecture is not scalable for a larger set of sensors covering a wider area of interest since the sensors are typically not capable of long distance communication. To allow the system to cope with additional load and to be able to cover a large area of interest without degrading the service, *network clustering* has been followed in some routing approaches.

The main aim of hierarchical routing is to efficiently maintain the energy consumption of sensor nodes by involving them in multi-hop communication within a particular cluster and by performing data aggregation and fusion in order to decrease the number of transmitted messages to the sink. Cluster formation is typically based on the energy reserve of sensors and sensor's proximity to the cluster head [52] [53]. LEACH [54] is one of the first hierarchical routing approaches for sensors networks. The idea proposed in LEACH has been an inspiration for many hierarchical routing protocols. Some of the hierarchical protocols are the following:

- Low-Energy Adaptive Clustering Hierarchy (LEACH) [54]
- Power-Efficient GAthering in Sensor Information Systems (PEGASIS) and Hierarchical-PEGASIS [55][56]
- Threshold sensitive Energy Efficient sensor Network protocol (TEEN) and Adaptive TEEN (APTEEN) [19] [57]
- Energy-aware routing for cluster-based sensor networks: Younis et al. [58]
- Self-organizing protocol: Subramanian et al. [59]

Since most of the hierarchical routing protocols are based LEACH, a brief overview of this protocol and PEGASIS are discussed. A detailed explanation of LEACH is presented in Section 2.4.

I. **Low-Energy Adaptive Clustering Hierarchy (LEACH)** [54]

It is one of the most popular hierarchical routing algorithms for sensor networks. The idea is to form clusters of sensor nodes based on the received signal strength and use local cluster heads as routers to the sink. This will save energy since the transmissions will only be done by such cluster heads rather than all sensor nodes. Optimal number of cluster heads is estimated to be 5% of the total number of nodes.

All the data processing such as data fusion and aggregation are local to the cluster. Cluster heads change randomly over time in order to balance the energy dissipation of nodes. The overall protocol architecture of LEACH is discussed thoroughly in Section 2.4.

II. **Power-Efficient GAthering in Sensor Information Systems (PEGASIS)** [55]

It is an improvement of the LEACH protocol. Rather than forming multiple clusters, PEGASIS forms chains from sensor nodes so that each node transmits and receives from a neighbor and only one node is selected from that chain to transmit to the base station (sink). Gathered data moves from node to node, aggregated and eventually sent to the base station. The chain construction is performed in a greedy way. As shown in Figure 2-8 node $c0$ passes its data to node $c1$. Node $c1$ aggregates node $c0$'s data with its own and then transmits to the leader $c2$. After node $c2$ passes the turn to node $c4$, node $c4$ transmits its data to node $c3$. Node $c3$ aggregates node $c4$'s data with its own and then transmits to the leader. Node $c2$ waits to receive data from both neighbors and then aggregates its data with its neighbors' data. Finally, node $c2$ transmits one message to the base station. The difference from LEACH is to use multi-hop routing by forming chains and selecting only one node to transmit to the base station instead of using multiple nodes. PEGASIS has been shown to outperform LEACH by about 100 to 300% for different network sizes and topologies. Such performance gain is achieved through the elimination of the overhead caused by dynamic cluster formation in LEACH and through decreasing the number of transmissions and reception by using data aggregation. However, PEGASIS introduces excessive delay for distant node on the chain. In addition the single leader can become a bottleneck

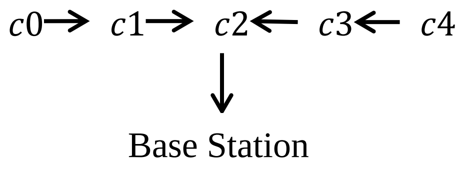


Figure 2-8: Chaining in PEGASIS [55]

2.4. Low-Energy Adaptive Clustering Hierarchy

LEACH is a self-organizing, adaptive clustering protocol that uses randomization to distribute the energy load evenly among the sensors in the network. In LEACH, the nodes organize themselves into local clusters, with one node acting as the local base station or cluster head. If the cluster heads were chosen a priori and fixed throughout the system lifetime, as in conventional clustering algorithms, it is easy to see that the unlucky sensors chosen to be cluster heads would die quickly, ending the useful lifetime of all nodes belonging to those clusters. Thus LEACH includes randomized rotation of the high-energy cluster head position such that it rotates among the various sensors in order to not drain the battery of a single sensor. In addition, LEACH performs local data fusion to “compress” the amount of data being sent from the clusters to the base station, further reducing energy dissipation and enhancing system lifetime [54].

Sensors elect themselves to be local cluster heads at any given time with a certain probability. These cluster head nodes broadcast their status to the other sensors in the network. Each sensor node determines to which cluster it wants to belong by choosing the cluster head that requires the minimum communication energy. Once all the nodes are organized into clusters, each cluster head creates a schedule for the nodes in its cluster. This allows the radio components of each non-cluster head node to be turned off at all times except during its transmit time, thus minimizing the energy dissipated in the individual sensor. Once the cluster head has all the data from the nodes in its cluster, the cluster head node aggregates the data and then transmits the compressed data to the base station. Since the base station is assumed to be far away, this is a high energy transmission. However, since there are only a few cluster heads, this only affects a small number of nodes.

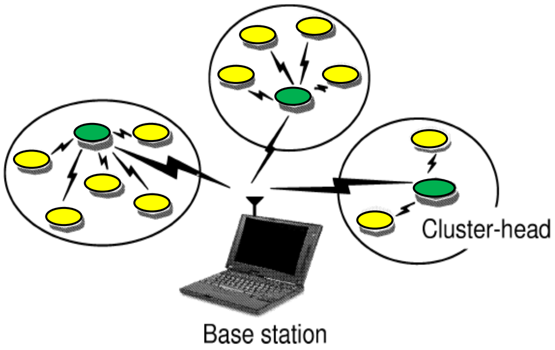
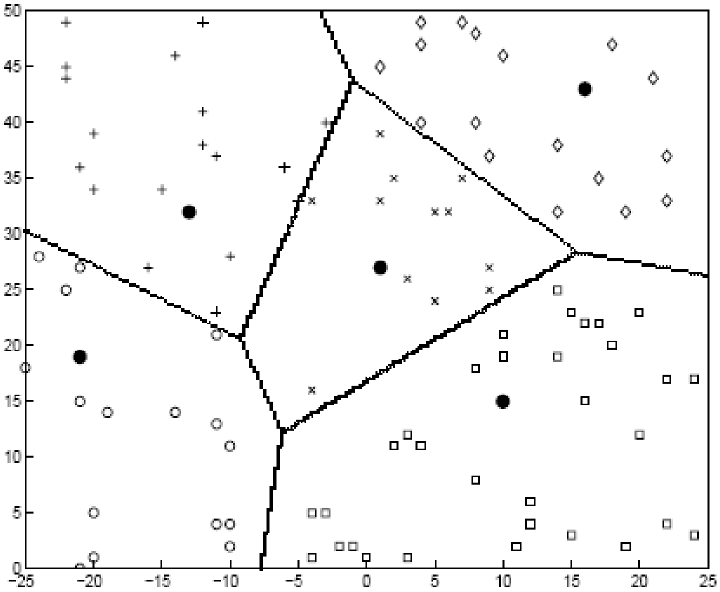
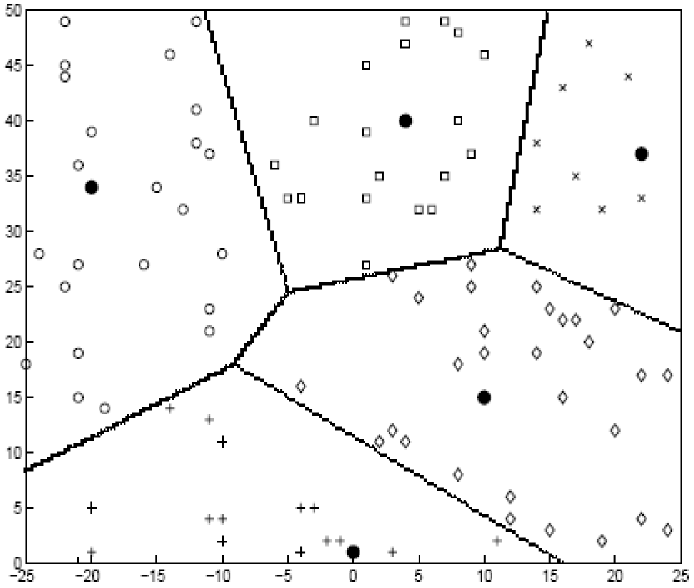


Figure 2-9: The LEACH protocol for microsensor networks [60]



(a)



(b)

Figure 2-10: Dynamic clusters: a) Cluster head nodes= $\mathbf{C}$ at time $\mathbf{t_1}$ (b) Cluster head nodes = $\mathbf{C'}$ at time $\mathbf{t_1 + d}$ [60]

As discussed previously, being a cluster head drains the battery of that node. In order to spread this energy usage over multiple nodes, the cluster head nodes are not fixed; rather, this position is self-elected at different time intervals. Thus a set C of nodes might elect themselves cluster heads at timet_1, but at time $\mathrm{t}_1 + \mathrm{d}$ a new set C' of nodes elect themselves as cluster heads, as shown in Figure 2-10.

LEACH’s operation is broken up into rounds, where each round begins with a ***setup phase***, when the clusters are organized, followed by a ***steady-state phase***, when data transfers to the base station occur. In order to minimize overhead, the steady-state phase is long compared to the set-up phase, as shown in Figure 2-11.

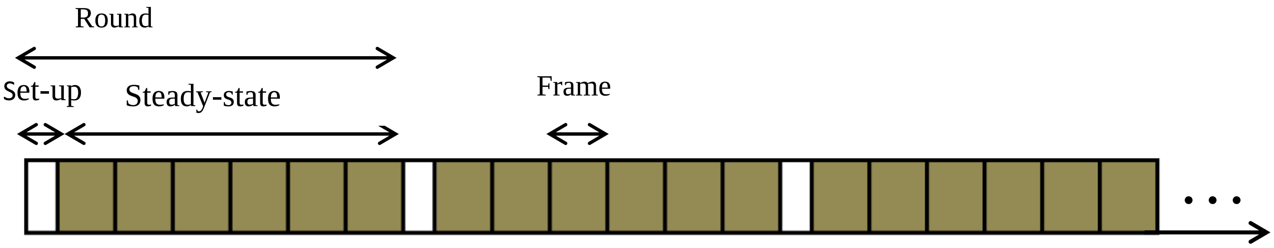


Figure 2-11: Time line showing LEACH operation [60]

I. **Setup phase**: includes advertisement, cluster set-up and schedule creation phases as shown in Figure 2-12.

a) Advertisement

Initially, when clusters are being created, each node decides whether or not to become a cluster head for the current round. This decision is based on the suggested percentage of cluster heads for the network (determined a priori) and the number of times the node has been a cluster head so far. This decision is made by the node i choosing a random number between 0 and 1. If the number is less than a threshold $T(i)$, the node becomes a cluster head for the current round r. The threshold is set as [54]:

$$T(i) = \begin{cases} \dfrac{p}{1 - p * (r \, mod \frac{1}{p})} & if \;\; i \in G \\ 0 & otherwise \end{cases} \qquad (2\text{-}3)$$

Where p is the percentage of cluster heads (e.g. 0.05), r is the current round, and G is the set of nodes that have not been cluster heads in the last $^1/_p$ rounds.

Using this threshold, each node will be a cluster head at some point within $1/p$ rounds. During round 0 $(\mathrm{r} = 0)$, each node has a probability p of becoming a cluster head. The nodes that are cluster heads in round 0 cannot be cluster heads for the next $1/p$ rounds. Thus the probability that the remaining nodes are cluster heads must be increased, since there are fewer nodes that are eligible to become cluster heads. After $1/p \; - 1$ rounds, $\mathrm{T} = 1$ for any nodes that have not yet been cluster heads, and after $1/p$ rounds, all nodes are once again eligible to become cluster heads.

Each node that has elected itself a cluster head for the current round broadcasts an advertisement message to the rest of the nodes. For this “cluster head-advertisement” phase, cluster heads use a CSMA MAC protocol, and all cluster head nodes transmit their advertisement using the same transmission energy. The non-cluster head nodes must keep their receivers on during this phase of set-up to hear the advertisements of all the cluster head nodes. After this phase is complete, each non-cluster head node decides the cluster to which it will belong for this round. This decision is based on the received signal strength of the advertisement. Assuming symmetric propagation channels, the cluster head advertisement heard with the largest signal strength is the cluster head to whom the minimum amount of transmitted energy is needed for communication. In the case of ties, a random cluster head is chosen [54].

b) Cluster Set-Up

After each node has decided to which cluster it belongs, it must inform the cluster head node that it will be a member of the cluster. Each node transmits this information back to the cluster head again using a CSMA MAC protocol. During this phase, all cluster head nodes must keep their receivers on.

c) Schedule Creation

The cluster head node receives all messages for nodes that would like to be included in the cluster. Based on the number of nodes in the cluster, the cluster head node creates a TDMA schedule telling each node when it can transmit. This schedule is broadcast back to the nodes in the cluster.

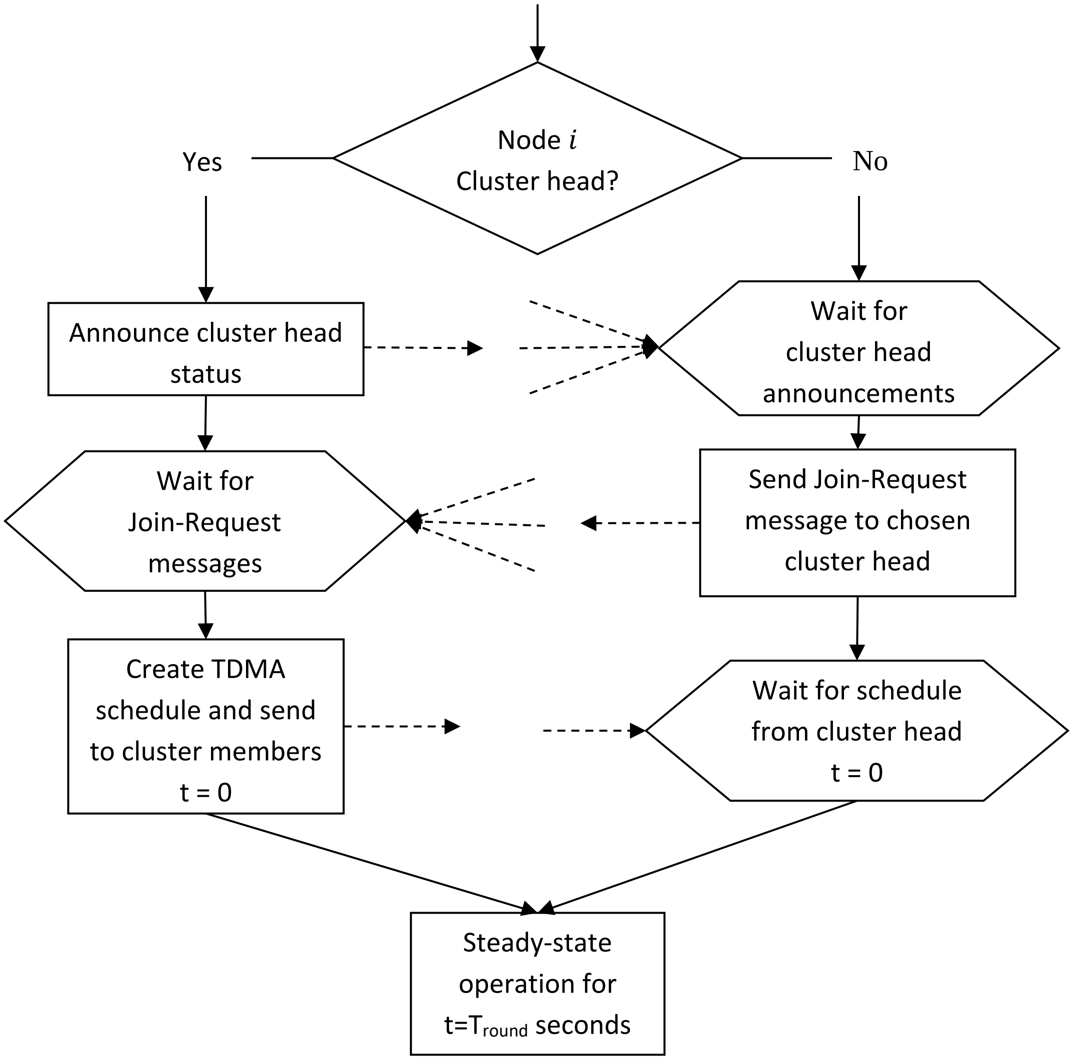


Figure 2-12: Flow-graph of the distributed cluster formation algorithm for LEACH [60]

II. Steady-State Phase (Data Transmission)

Once the clusters are created and the TDMA schedule is fixed, data transmission can begin. Assuming nodes always have data to send, they send it during their allocated transmission time to the cluster head. This transmission uses a minimal amount of energy (chosen based on the received strength of the cluster head advertisement). The radio of each non-cluster head node can be turned off until the node's allocated transmission time, thus minimizing energy dissipation in these nodes. The cluster head node must keep its receiver on to receive all the data from the nodes in the cluster. When all the data has been received, the cluster head node performs signal processing functions to compress the data into a single signal. For example, if the data are audio or seismic signals, the cluster head node can beamform the individual signals to generate a composite signal. This composite signal is sent to the base station. Since the base station is far away, this is a high-energy transmission.

This is the steady-state operation of LEACH networks, as shown in Figure 2-13. After a certain time, which is determined a priori, the next round begins with each node determining if it should be a cluster head for this round and advertising this information, as described above.

III. Multiple Clusters

The preceding discussion describes how the individual clusters communicate among nodes in that cluster. However, radio is inherently a broadcast medium. As such, transmission in one cluster will affect (and hence degrade) communication in a nearby cluster. For example, Figure 2-14 shows the range of communication for a radio. Node A's transmission, while intended for Node B, corrupts any transmission to Node C. To reduce this type of interference, each cluster communicates using different CDMA codes. Thus, when a node decides to become a cluster head, it chooses randomly from a list of spreading codes. It informs all the nodes in the cluster to transmit using this spreading code. The cluster head then filters all received energy using the given spreading code. Thus neighboring clusters' radio signals will be filtered out and not corrupt the transmission of nodes in the cluster.

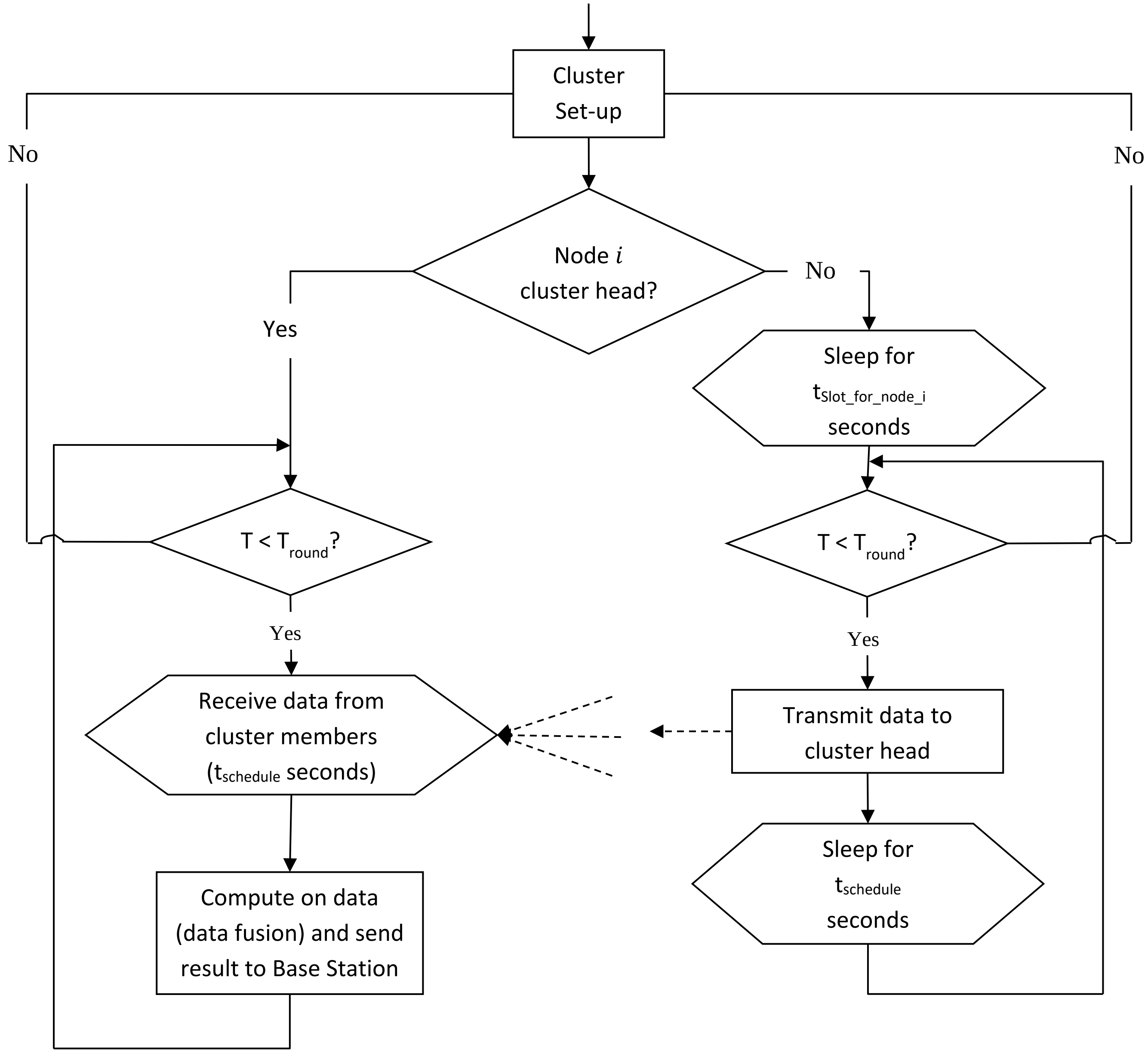


Figure 2-13: Flow-graph of the steady-state operation for LEACH [60]

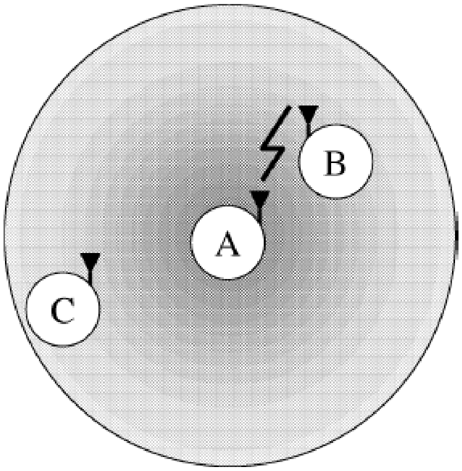


Figure 2-14: Radio interference. Node A’s transmission to node B corrupts any transmission to node C [54].

LEACH achieves over 7 times reduction in energy dissipation compared to direct communication and 4-8 times compared to the minimum transmission energy routing protocol. The nodes die randomly and dynamic clustering increases lifetime of the system. LEACH is completely distributed and requires no global knowledge of network. However, LEACH uses single-hop routing where each node can transmit directly to the cluster head and the sink. Therefore, it is not applicable to networks deployed in large regions. Furthermore, the idea of dynamic clustering brings extra overhead, e.g. head changes, advertisements etc., which may diminish the gain in energy consumption [54].

Genetic Algorithm

Genetic Algorithm (GA), which is introduced in the 1970s by John Holland at University of Michigan, is a search heuristic that mimics the process of natural evolution [61]. This heuristic is routinely used to generate useful solutions to optimization and search problems. GAs belong to the larger class of evolutionary algorithms (EAs), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover.

In a GA, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached [62].

To use a GA, we must encode solutions (chromosomes) to our problem in a structure that can be stored in the computer. The GA creates a population of genomes then applies crossover and mutation to the individuals in the population to generate new individuals. It uses various selection criteria so that it picks the best individuals for mating (crossover). Our objective function determines how good each individual is.

A typical GA requires [63]:

- A genetic representation of the solution domain.
- A fitness function to evaluate the solution domain.

A standard representation of the solution is as an array of bits. Arrays of other types and structures can be used in essentially the same way. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, which facilitates simple crossover operations.

The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. For instance, in the clustering of wireless sensor nodes, one wants to minimize the total energy consumption of the network. A representation of a solution might be an array of bits, where each bit represents a wireless node, and the value of the bit (0 or 1) represents whether or not the wireless sensor node is a cluster head.

GAs have applications in bioinformatics, computational science, engineering, economics, chemistry, manufacturing, mathematics, physics and other fields.

3.1. Operators of Genetic Algorithm

In the following section, different technical terms and operators that are used in the GA are explained.

I. Population

A population consists of a group of individuals called chromosomes that represent a complete solution to a defined problem. Each chromosome is a sequence of 0s or 1s. The initial set of the population is a randomly generated set of individuals. A new population is generated by two methods: steady-state GA and generational GA. The steady-state GA replaces one or two members of the population; whereas the generational GA replaces all of them at each generation of evolution [62].

II. Fitness

In nature, an individual's fitness is its ability to pass on its genetic material. This ability includes traits that enable it to survive and further reproduce. In a GA, fitness is evaluated by the function defining the problem. The fate of an individual chromosome depends on the fitness value. The chances of survival are higher for better fitness values.

Once we have the genetic representation and the fitness function defined, GA proceeds to initialize a population of solutions randomly, and then improve it through repetitive application of mutation, crossover, and selection operators.

III. Initialization

Initially many individual solutions are randomly generated to form an initial population. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Traditionally, the population is generated randomly, covering the entire range of possible solutions (the search space). Occasionally, the solutions may be selected in areas where optimal solutions are likely to be found.

IV. Selection

The selection process determines which of the chromosomes from the current population will mate (crossover) to create new chromosomes. The chromosomes are selected through a fitness-based process, where fitter chromosomes (as measured by a fitness function) are typically more likely to be selected. These new chromosomes join the existing population. This combined population will be the basis for the next selection. The individuals (chromosomes) with better fitness values have better chances of selection. There are several selection methods, such as: *Roulette-Wheel selection*, *Rank selection* and *Tournament selection*.

a) Roulette-Wheel (Fitness Proportionate) selection

The fitness level, which is assigned to possible solutions or chromosomes by the fitness function, is used to associate a probability of selection with each individual chromosome. If f_i is the fitness of individual i in the population, its probability of being selected is

$$ps_i = \frac{f_i}{\sum_{j=1}^{P} f_j} \tag{3-1}$$

Where P is the number of individuals in the population (population size) [62].

b) Tournament selection

It involves running several "tournaments" among a few individuals chosen at random from the population. The winner of each tournament that is the one with the best fitness is selected for crossover. If the tournament size is larger, weak individuals have a smaller chance to be selected.

c) Rank selection

It picks the best individuals from the population every time. There are other selection methods that use the combination of the above selection schemes and keep the fittest individual.

V. Reproduction

The next step is to generate a second generation population of solutions from those selected through genetic operators: crossover (also called recombination), and/or mutation.

a) Crossover

Crossover is also known as recombination of component materials due to mating. It is a simulation of the sexual reproductive process which is responsible for the transfer of genetic inheritance. The outcome of crossover heavily depends on the selection of chromosomes made from the population. Crossover is a binary genetic operator acting on two parents. There are different crossover operators that have been developed for various purposes. The simplest is the single-point crossover whereby a point is chosen at random, and the two parent chromosomes exchange information after that point. An example of single-point crossover is shown in Figure 3-1.

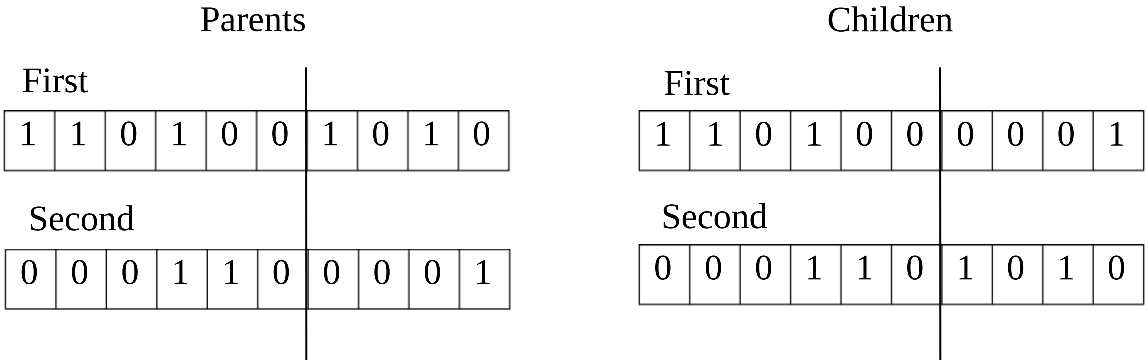


Figure 3-1: A single-point crossover

In Figure 3-1 the bit sequences of first chromosome, starting from the crossover point, are copied to the second chromosome and vice versa. The crossover results in two new offspring that have different bit sequence from their parents. Crossover is done after the selection process and depends on the probability defined for the crossover called crossover rate. The probability that the crossover will take place depends on the crossover rate. Generally the crossover rate is high, around 80 to 95 percent [64].

b) Mutation

As a result of crossover, the new generation introduced will only have the traits of the parents. This can sometimes lead to a problem where no new genetic material is introduced in the offspring. Mutation allows new genetic patterns to be introduced in the new chromosomes. Mutation introduces a new sequence of genes into a chromosome but there is no guarantee that mutation will produce desirable features in the new chromosome. The selection process will retain it if the fitness of the mutated chromosome is higher than the general population, otherwise, selection will ensure that the chromosome does not live to mate in future. As with crossover, the mutation rate is defined to control how often mutation is applied. Unlike crossover, the mutation rate is very low, around 0.5 to 1 percent.

In GAs, the probability of mutation can be implemented either on a per-bit basis or on a per-chromosome basis. For a per-bit basis, if the mutation rate is 0.001, each bit in a chromosome has 0.1 percent chance of being mutated. For a per-chromosome basis, the mutation rate of 0.001 means there is a 0.1 percent chance of a chromosome being mutated [64].

In this thesis, the proposed GA implements mutation on a per-bit basis. Mutation, unlike crossover, is a unary genetic operator that affects only a single chromosome. A chromosome selected for mutation will have a randomly selected bit changed from 0 to 1, or vice versa. An example of mutation is shown in Figure 3-2.

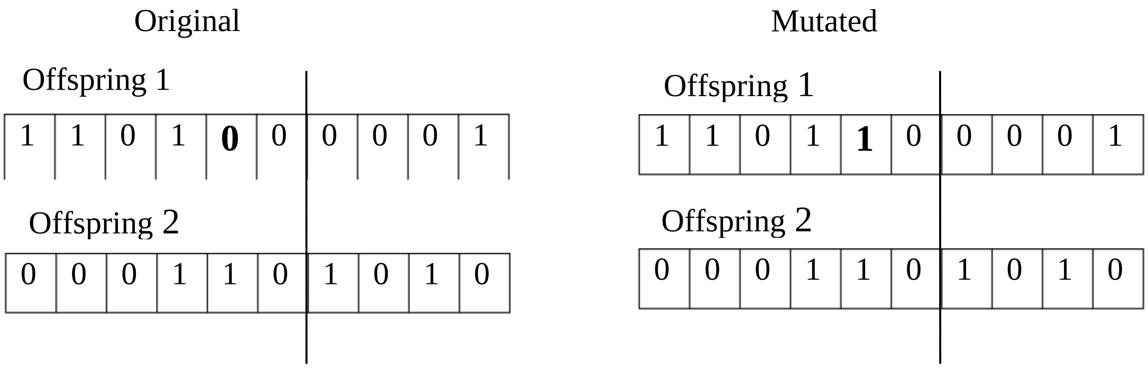


Figure 3-2: An example of Mutation

Figure 3-2 shows the effect of mutation on the two offspring created as a result of crossover. During mutation, the fifth bit of offspring 1 is changed from0 to 1; however, due to very low probability of mutation, there is no mutation in offspring 2.

The above processes ultimately result in the next generation population of chromosomes that is different from the initial generation. Generally the average fitness will have increased by this procedure for the population, since only the best organisms from the first generation are selected for breeding, along with a small proportion of less fit solutions.

Although Crossover and Mutation are known as the main genetic operators, it is possible to use other operators such as regrouping, colonization-extinction, or migration in GAs.

VI. Termination

This generational process is repeated until a termination condition has been reached. Common terminating conditions are:

- A solution is found that satisfies minimum criteria
- Fixed number of generations reached
- Allocated budget (computation time/money) reached
- The highest ranking solution's fitness is reaching or has reached a stable state such that successive iterations no longer produce better results
- Manual inspection
- Combinations of the above

The following algorithm describes the overall operation of GA.

3.2. Basic Genetic Algorithm

```
{       % Generate random population of chromosomes
        Initialize population;
        % Evaluate the fitness of each chromosome in the population
        Evaluate population;                [Fitness]
        % Create, accept, and test a new population:
        while Termination_Criteria_Not_Satisfied
        {       % Select according to fitness
                Select parents for reproduction;    [Selection]
                % With a crossover probability perform crossover or copy parents
                Perform crossover;                  [Crossover]
                % With a mutation probability mutate offspring at each position in chromosome
                Perform mutation;                   [Mutation]
```

```
        Accept new generation;
        Evaluate population;            [Fitness]
    }
}
```

Design and Implementation

In this chapter, the design and implementation of the proposed routing protocols, namely Genetic Algorithm Inspired Clustering Hierarchy (GAICH) and Grid Clustering Hierarchy (GCH), are presented. In the first section, the protocol architecture of GAICH is discussed in detail. Then, GCH is designed and implemented.

In the design and analysis of these protocols, the following scenarios/ assumptions are taken into consideration:

- The sensor nodes are deployed in a field randomly with uniform distribution.
- The base station is fixed somewhere near the center of the network.
- The base station is a super node that has no limitations in computation, communication and energy capacity.
- All the nodes in the network have the same initial battery power and their computation, communication and energy capacity is limited.
- The base station has location information about the sensor nodes after they are deployed in the network. Since the base station has high computation capability, it can determine the location of a sensor node in the network by applying localization algorithms. It is also possible to make the sensor nodes GPS enabled then each node will have location information. But it is too expensive to include a GPS receiver in every node of the sensor network.

In this work, LEACH, which is discussed in the Chapter 2, is implemented with variable number of clusters on different base station positions from the center of the network. The simulation results of LEACH are compared with the proposed clustering algorithms: GAICH and GCH.

4.1. Genetic Algorithm Inspired Clustering Hierarchy

In this protocol, a GA is used at the base station to determine the optimum number of cluster heads that create energy efficient clusters for a given number of transmissions. The sensor node is represented as a bit of a chromosome. The cluster head and cluster member nodes are represented as 1s and 0s respectively. A population consisting of several chromosomes is generated with a certain percentage of the nodes as cluster heads. The best chromosomes are used to generate the next population. Based on the survival fitness, the population transforms into the future generation. Initially, each fitness parameter is assigned an arbitrary weight; however, after every generation, the fitness of each chromosome is evaluated and the weights for each fitness parameter are updated accordingly.

The GA outcome identifies best cluster heads for the network. The base station broadcasts the outcomes of the GA to the sensor nodes in the network. These messages may include: the number of cluster heads and their corresponding identification number, the number of transmissions for this configuration and other important information. All the sensor nodes receive the packets broadcasted by the BS and cluster heads broadcast their state of being cluster heads to the other nodes. The nodes that are not cluster heads and heard these broadcasts decide to join their nearest cluster head nodes to form the clusters. This phase is the same as LEACH and completes the cluster formation phase. Then, the data transfer phase follows.

To apply the GA that is discussed in Chapter 2, a genetic representation to the problem and fitness function to evaluate the solution domain should be defined.

I. Chromosome Representation

The chromosomes are represented by two linear arrays; the first one contains sequences of 0s and 1s that represent the sensor nodes in the WSN where 1s are cluster heads and 0s are cluster members. The second array contains the identification numbers (index) of the nodes corresponding to the first one. The size of the array is equal to the currently alive nodes. For example, this type of representation is shown in the following table.

value	0	0	0	1	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	1
index	5	7	8	9	12	11	14	15	16	20	21	22	23	30	40	41	42	45	47	49

Table 4-1: Chromosome representation with two dimensional arrays with the first row representing the role of the sensor nodes and the second containing the corresponding node IDs.

II. Fitness Function

The fitness function is defined over the above genetic representation and measures the quality of the represented solution. In this work, the fitness function should provide the means to determine the optimum number of cluster heads that create energy efficient clusters for a given number of transmissions. The fitness function will take into consideration different parameters that, directly or indirectly, affect the energy consumption and network lifetime of the WSN.

The fitness of a chromosome is designed to minimize the energy consumption and extend the network lifetime of the network. The fitness parameters that are considered in this work are described as follows:

1) *Sum of Cluster head Distances to Base Station (SCHDBS):* It is the sum of all cluster head distances from the base station and defined as follows:

$$\text{SCHDBS} = \sum_{i=1}^{L} d_{iBS} \tag{4-1}$$

   Where L is the number of cluster heads and d_{iBS} is the distance between cluster head i and the base station [65].

2) *Sum of Cluster heads Density (SCHD):* The density of a node is defined as the number of nodes around it where their distances are less than a threshold distance d_{th}, from the node. SCHD is the sum of density parameters of the cluster heads. If the value of a node’s density is greater, it will be a better candidate for the cluster head election. As a result, nodes around this candidate node will consume minimum energy to transmit their data to it. So the greater the SCHD value means the better the chromosome.

$$\mathrm{SCHD} = \sum_{i=1}^{L} \rho_i \tag{4-2}$$

Where L is the number of cluster heads and ρ_i is the density of the cluster head i [65].

3) *Sum of the Cluster heads Centrality (SCHC):* The centrality of a node is defined as the inverse of the sum of the squared distances of the other nodes that are within the threshold distance d_{th} from it. It describes how central is a node to a cluster. Since transmission energy is proportional to distance square, the higher the value of the centrality of a node, the lower the amount of energy required by the other nodes to transmit their data through this node as cluster head.

   When several neighbor nodes have the same density parameter, the node at the center is the best candidate to be the cluster head. SCHC is the sum of centrality of cluster heads.

$$\mathrm{SCHC} = \sum_{i=1}^{L} Cnt_i$$

$$Cnt_i = \frac{1}{\sum_{j=1}^{n} {d_{ij}}^2} \tag{4-3}$$

   Where L is the number of cluster heads, Cnt_i is the centrality of the cluster head i , n is the number of nodes within the threshold distance d_{th}, and d_{ij} is the distance from the cluster head i to node j [64] [65].

4) *Sum of the Cluster heads Residual Energy (SCHRE):* As the cluster heads are responsible for the reception and transmission of more data packets than member (normal) nodes, those nodes with a greater residual energy are better candidates for the cluster head election. SCHRE is defined as the sum of the residual energy of the cluster heads.

$$\mathrm{SCHRE} = \sum_{i=1}^{L} re_i \tag{4-4}$$

Where L is the number of cluster heads and re_i is the residual energy of the cluster head i [65].

The chromosome fitness is defined as a function of all the above fitness parameters [64], i.e.

$$F = \sum_i (w_i * f_i) \;\; \forall f_i \in \{SCHDBS, SCHD, SCHC, SCHRE\}$$

$$F = w_1 * \left(\frac{1}{SCHDBS}\right) + w_2 * SCHD + w_3 * SCHC + w_4 * SCHRE \tag{4-5}$$

Where the initial fitness parameters are assigned with arbitrary initial weights, w_i. then, after every generation the fitness value of the best chromosome is evaluated and the weights for fitness parameters are updated as follows:

$$w_i = w_i' + c_i \Delta f_i$$

$$\Delta f_i = f_i - f_i' \tag{4-6}$$

Where w_i and w_i' are the current and previous weights, f_i and f_i' are the fitness parameter values of the current and previous best chromosomes respectively, the coefficient $c_i = 1/(1 + e^{-f_i'})$ improves the value of the weights based on the previous experience [66]. A suitable range of initial weights can be determined during simulation.

As discussed above, the genetic representation and the fitness function for the goal of minimizing the energy consumption and extending the network life time of the network is defined. The GA then proceeds to initialize a population of solutions randomly and improves it through repetitive application of selection, crossover, and mutation operators until the maximum number of generations is reached.

III. Initialization (Population Generation)

The initialization step in most GA applications is made as much as simple. Generally, GAs start with an initial population that is generated in random. In this work, a simple procedure to generate the chromosomes for the initial population is proposed. As stated above, the WSN nodes are represented as bits of a chromosome where cluster head and member nodes are represented as 1s and 0s, respectively. The method uses the average energy of the network into considerations to determine the percentage of cluster heads in a chromosome. In addition, a node is considered to be a cluster head if its energy level is more than the average energy of the network. In this way, the method can create higher quality initial population. The current average energy of the network is defined as:

$$Ave_E = \sum_i^n E_i \big/ n \qquad (4\text{-}7)$$

Where E_i is the current energy level of node i and n is the number of currently available nodes.

Here, three levels of cluster head percentages that depend on the current average energy (Ave_E) of the network are taken. The first level is taken when Ave_E is above half of the initial average energy of the network which is equal to the initial energy of an individual node since I take a homogenous WSN. The second level is used when $\mathrm{Ave_E}$ is between half and one third of the initial average energy of the network. The third level is taken otherwise. The percentage of cluster heads, p, is defined as follows:

$$p(\mathrm{Ave_E}) = \begin{cases} 1^{st}\ level, & Ave_E \geq Ave_{E_i}/2 \\ 2^{nd}\ level, & Ave_{E_i}/2 < Ave_E \geq Ave_{E_i}/3 \\ 3^{rd}\ level, & otherwise \end{cases} \qquad (4\text{-}8)$$

Where 1^{st} , 2^{nd} and 3^{rd} levels are percentage of cluster heads, p, that depend on the current, Ave_E, and initial, Ave_{E_i}, average energy of the network. These levels can be set depending on the simulation setup. For instance, 1^{st}, 2^{nd} and 3^{rd} levels can be set to 0.15, 0.10, and 0.5 respectively.

Once the percentage of cluster heads is determined for a chromosome, nodes where their energy level is more than the current average energy of the network are chosen randomly to be cluster heads. That means in the chromosome the corresponding nodes will be represented by 1's. The probability of node i to be part of a chromosome is

$$\gamma\,(i) = \begin{cases} 1, & E_i \geq Ave_E \\ 0, & E_i < Ave_E \end{cases} \tag{4-9}$$

Where E_i is the current energy level of node i and Ave_E is the current average energy of the network.

Using the above procedure, the system generates a certain number of chromosomes equal to the population size. In most of the GA applications, the population size is constant during the evolutional search. In this work, the population size is dependent on the percentage of cluster heads and the number of eligible nodes for cluster heads as stated in the above method. So the population size parameter of the GA will be variable. The population size $, P,$ is defined as follows:

$$P = \frac{Total\ number\ of\ eligible\ Nodes}{p} \tag{4-10}$$

Where a node is eligible when its γ is 1 and p is the percentage of cluster heads.

Once the population size, percentage of cluster heads and set of eligible nodes for a cluster head are determined, the chromosomes for the first generation will be generated.

IV. Fitness Evaluation

After the initial population is generated, the fittest value of each individual (chromosome) is evaluated. The fitness function that is defined above is used here.

V. Selection

The selection process determines which of the chromosomes from the current population will mate to create new chromosomes. There are several selection methods as discussed in Chapter 2.

In this work, Roulette-Wheel (Fitness Proportionate) selection method is applied. As stated in Chapter 2, If f_i is the fitness of individual (chromosome) i in the population, its probability of being selected is given by Equation 3-1.

The procedure generates a population of parents in the order of their selection. It runs for P iterations. This results in a parent population that contains the combination of best and poor fitness value individuals. Then, selected parents will be used to generate the next generation individuals by the genetic operators: crossover and mutation.

VI. Reproduction: *Crossover* and *Mutation*

The selected parents mate to produce new offspring. The procedure to crossover two parents is as follows. The first parent is selected from the parent population sequentially and a mate that is the second parent is selected from the parent population randomly. Crossover is made on the two parents. This runs for P iterations. From each iteration, it takes one of the two children to the next generation. As a result the children population size will be the same P. Here the probability of making a crossover between two selected parents is $pc = 0.8$.

After the crossover operation is finished, starting from the first child to the last, mutation is done to the child chromosomes with a mutation probability of $pm = 0.006$. As a result new genetic patterns will be introduced to the new generation.

VII. Termination

In this work, the GA runs until the maximum number of generations is reached. Here, the maximum number of generation is set to N generations by taking into consideration the cost of computation complexity and latency. The number can be increased to get better chromosomes that give as maximum fitness value. But as the maximum number of generations increase the computational complexity of the algorithm increases. Depending on the base station’s computation capacity which runs the GA, this number can be increased to get better chromosomes.

After the GA is completed, the chromosome that is the best from the last generation is taken to be the network configuration, as shown in Figure 4-1. This chromosome identifies the cluster heads that results in the minimum energy consumption and extended lifetime for the WSN.

The base station broadcasts this result to the network. Thus, nodes that are selected to be cluster heads broadcast their nomination to their neighbors. Then, the normal nodes decide to join clusters depending on the signal strength that is heard from the cluster heads. This phase of the energy efficient protocol is the same LEACH.

GAICH's operation is broken up into *rounds* like LEACH. The whole operation of GAICH is divided into two phases: *configuration and steady-state phases*, as shown in Figure 4-2. In the *configuration phase* the base station runs the GA and determines the cluster heads that give the best fitness values. Once the base station knows this, it will determine for how many rounds the current configuration last. The number of rounds , R_{C_i} , for this configuration can be fixed or made adaptive based on the fitness value of the current and previous configuration. The base station broadcasts to the network the cluster head information and number of rounds, R_{C_i} , for this configuration. Then, the clusters are created and TDMA schedule is fixed for nodes within the clusters.

In the *steady-state phase*, like LEACH sensed data is transmitted to the base station. Once the clusters are created and the TDMA schedule is fixed, data transmission can begin. Assuming nodes always have data to send, they send it during their allocated transmission time to the cluster head. When all the data has been received from nodes within the cluster, the cluster head performs signal processing functions to compress the data. The compressed data is sent to the base station. This phase lasts for R_{C_i} number of rounds. Then, the next *configuration phase* begins with the base station running the GA.

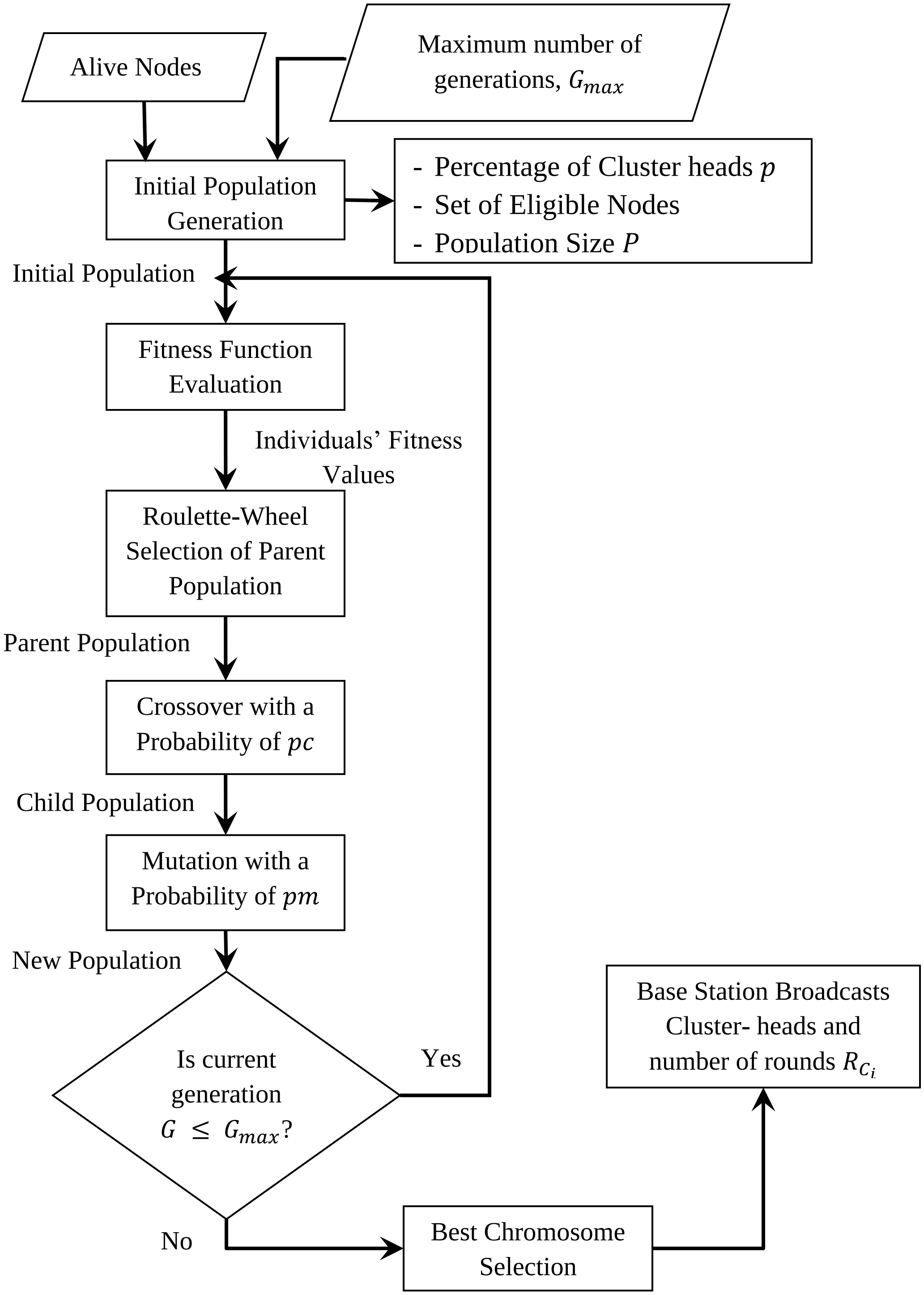


Figure 4-1: Flowchart of the Genetic Algorithm that runs on the Base Station. The outputs are the Cluster heads and the number of rounds $\mathbf{R_{C_i}}$ for the current configuration.

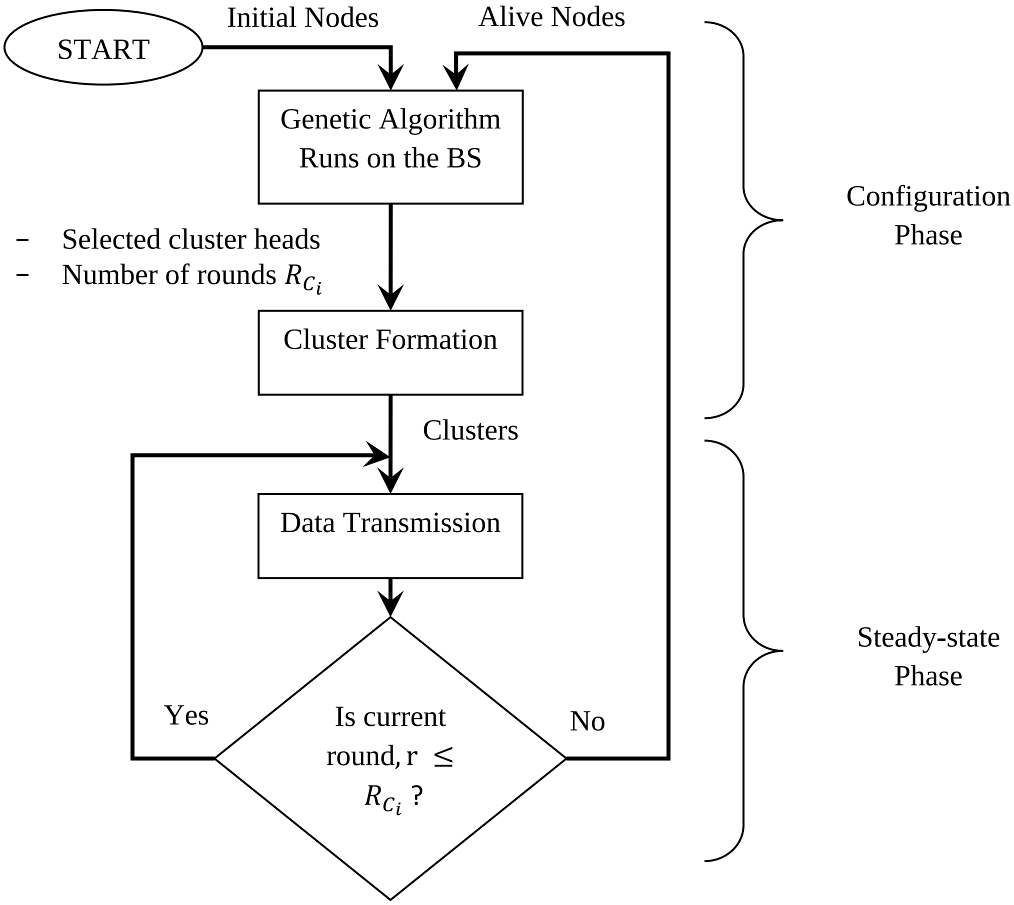


Figure 4-2: Flowchart of the operations of GAICH. Here the cluster formation and data transmission parts are the same as LEACH.

4.2. Grid Clustering Hierarchy

In this protocol, the base station divides the network into a variable number of virtual grids depending on the current average energy of the network. Once the nodes within the grids are known, a cluster head node is selected from each grid. In addition, the cluster head role rotates within the nodes in the grid in a round-robin manner. After a certain number of transmissions, the base station checks the average energy of the network. Then, if the average energy of the network is less than a certain threshold value determined a prior, the number of virtual grids will be changed and the above procedure continues.

In this work, a three level virtual grid formation is presented. The first one is when the average energy of the network is greater than two-third of the initial average energy of the network. In this level, 16 equally divided square grids are taken, as shown in Figure 4-3a. In the second level shown in Figure 4-3b, four equal square grids are taken when the average energy of the network is between two-third and one-third of the initial energy. Finally, the third level is taken when the average energy of the network is less than one-third of the initial energy and the whole network is consider as one square grid, as show in Figure 4-3c. The number of grids and levels can be made variable depending on the network size and the initial amount of energy of the nodes. Generally, the number of gird, g, is defined as follows:

$$g(n) = 2^{r*n-2} \tag{4-11}$$

Where n is the level number and r is rate of partitioning

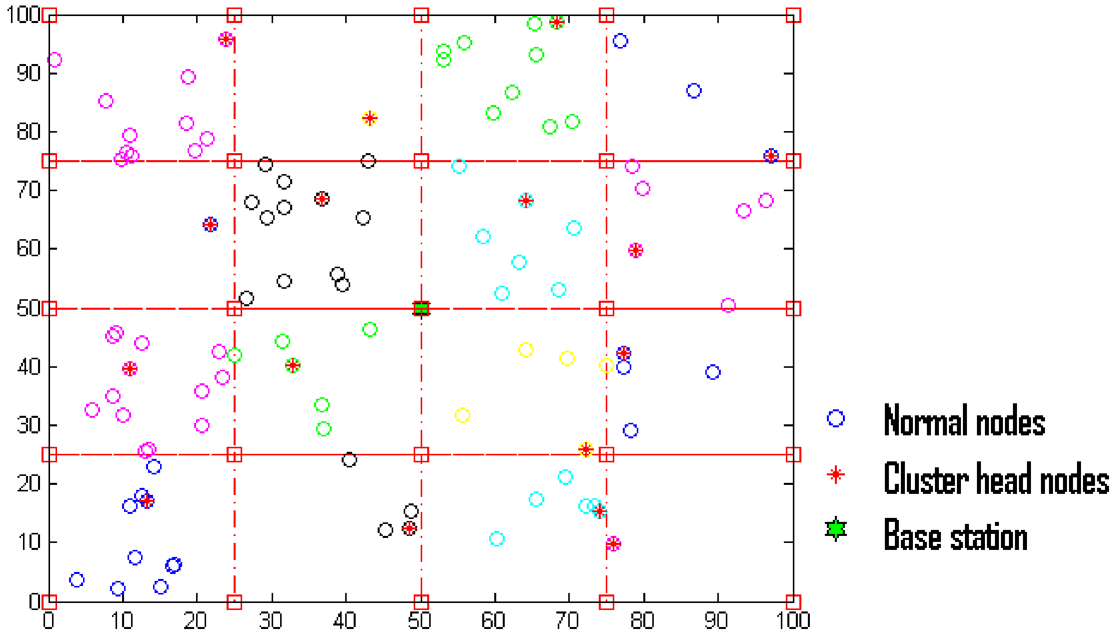
As three levels depending on the current, Ave_E, and initial, Ave_{E_i} , average energy of the network are taken, the number of gird, g, becomes:

$$g(Ave_E) = \begin{cases} 1^{nd}\ \ level, & Ave_E \geq Ave_{E_i} * 2/3 \\ 2^{nd}\ \ level, & Ave_{E_i} * 2/3 < \ Ave_E \geq \ Ave_{E_i}/3 \\ 3^{rd}\ \ level, & otherwise \end{cases} \tag{4-12}$$

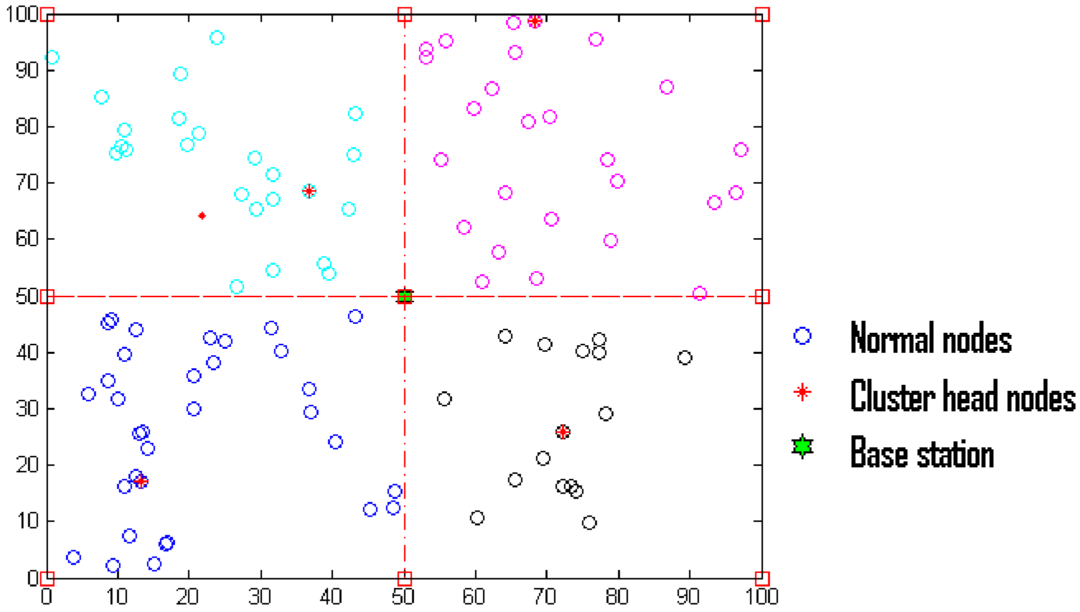
Where 1^{st},2^{nd} and 3^{rd} levels are the predetermined number of grids, g, i.e. 1^{st}, 2^{nd} and 3^{rd} levels are 16, 4, and 1 respectively, as shown in Figure 4-3.

Once the base station determines the nodes in the virtual grid, it forms a round-robin schedule for the cluster head role in each grid. This information is broadcasted by the BS to the sensor nodes in the network. These messages may include: the cluster head nodes, round-robin schedule in each grid, the number of transmissions with this configuration and other important information. All the sensor nodes receive the packets broadcasted by the base station and cluster head nodes broadcast their state of being cluster head to the other nodes. The nodes that are not cluster heads and heard these broadcasts decide to join their nearest cluster heads to form the clusters. This phase is the same as LEACH and completes the cluster formation phase. Then, the data transfer phase follows.

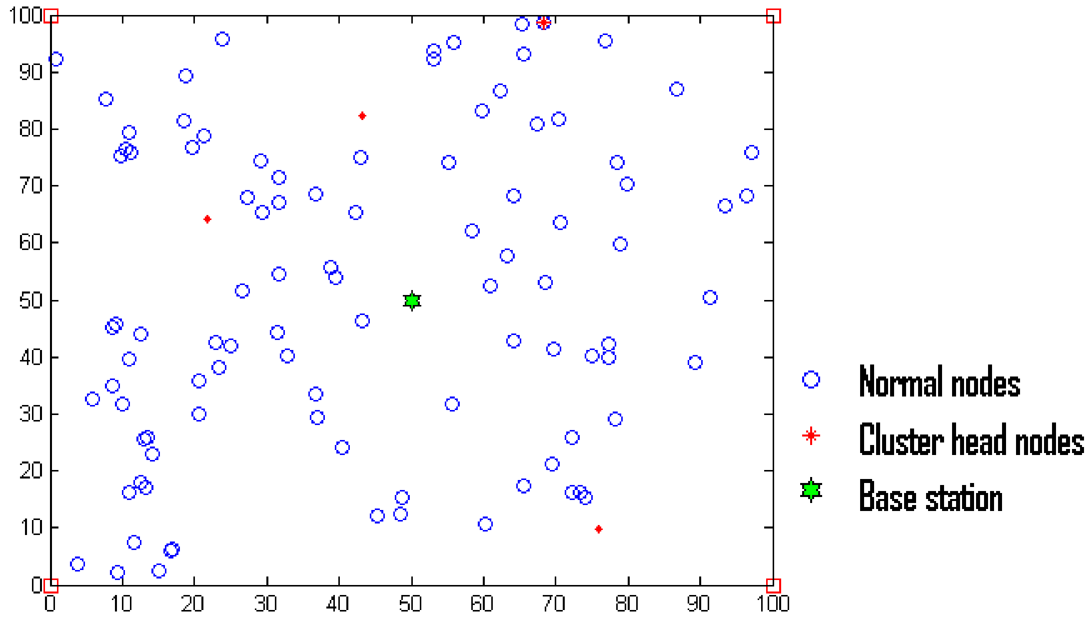
In the next round, based on the round-robin schedule the next node in the grid takes the cluster head role. Then, the cluster formation and data transfer phase continues again.



(a)



(b)



(c)

Figure 4-3: Three level virtual gridding in GCH for a 100-node random test network where the base station is located at the center of the network, at location (x=50, y=50). (a) First level gridding. (b) Second level gridding (c) Third level gridding

Generally, GCH's operation is broken up into *rounds* like LEACH. The whole operation of GCH is divided into *three* phases: *Gridding, cluster formation and steady-state phases*, as shown in Figure 4-4. In the g*ridding phase* the base station determines the number of virtual grids, the nodes in these grids and the round-robin schedule for the cluster head role. Once the base station determines this, it will determine for how many rounds the current grid level lasts. The number of rounds R_{C_i} , for this configuration can be fixed or made adaptive based on the average energy of the network. The base station broadcasts this information to the sensor nodes in the network. Then, the *cluster formation phase* follows by creating the clusters and TDMA schedule is fixed for nodes within the clusters.

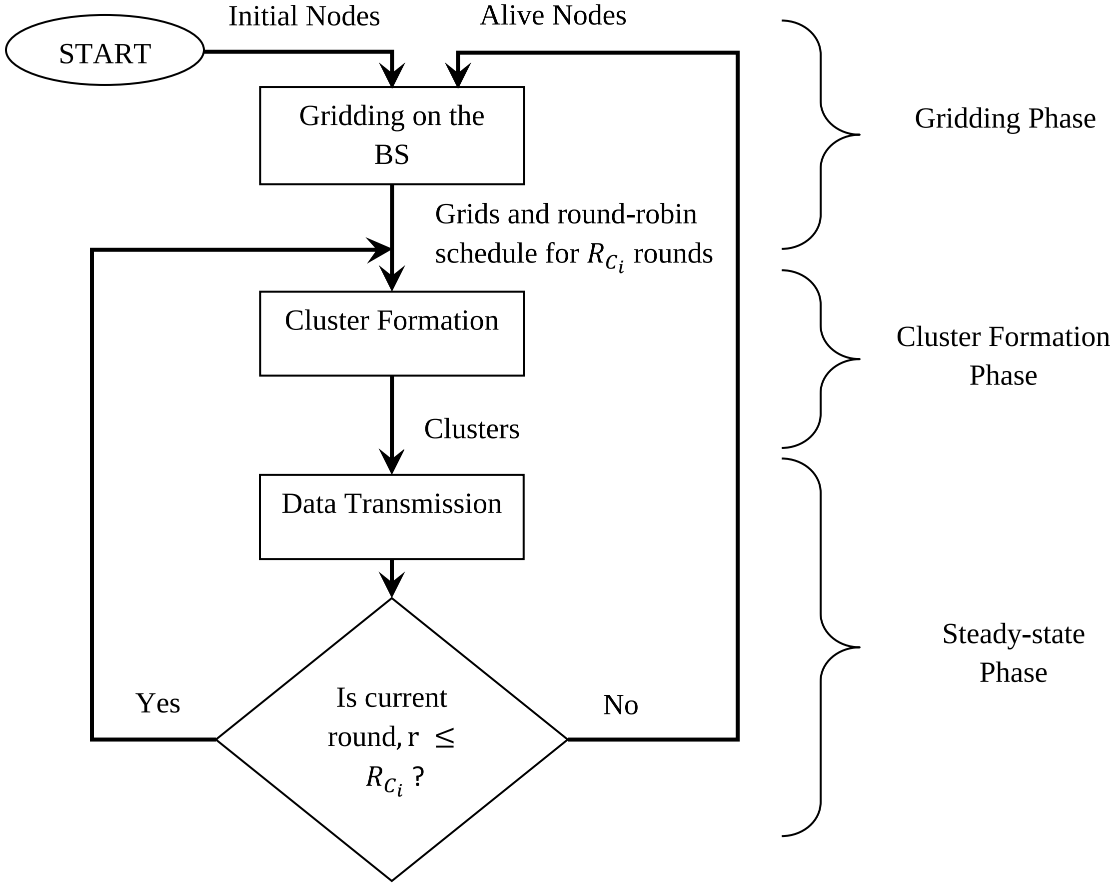


Figure 4-4: Flowchart of the operations of GCH.

In the *steady-state phase*, like LEACH sensed data is transmitted to the base station. Assuming nodes always have data to send, they send it during their allocated transmission time to the cluster head. When all the data has been received from nodes within the cluster, the cluster head performs signal processing functions to compress the data. The compressed data is sent to the base station. After one round, the cluster head role is changed to the next node in the grid based on the round-robin schedule. Then, the *cluster formation phase* continues by creating the clusters and fixing the TDMA schedule for nodes within the clusters. Again the Steady-state Phase follows after the Custer Formation Phase. The *cluster formation phase* and *steady-state phases* exchange lasts for R_{C_i} number of rounds. Then, the next *gridding phase* begins with the base station determining the level of gridding.

Simulation and Results

It is impossible to analytically model the interactions between all the nodes in the WSN even for a moderately-sized network with tens of nodes. Therefore, simulation is used to determine the benefits of the protocols that are discussed in Chapter 4. Computation and radio energy dissipation models are implemented in MATLAB to support the design and simulation of these protocol architectures.

In the test carried out in this chapter, the protocols GAICH and GCH that are proposed in Chapter 4 are compared with LEACH and direct transmission in terms of network lifetime, energy dissipation and amount of data transfer (raw data for direct transmission, aggregate data for the other protocols).

5.1. Simulation Models

In order to compare different protocols, it is important to have good models for all aspects of communication. The models for channel propagation, radio energy dissipation and beamforming computation energy are discussed below.

5.1.1. Channel Propagation Model

In a wireless channel, the electromagnetic wave propagation can be modeled as a power law function of the distance between the transmitter and receiver. In addition, if there is no direct line-of-sight path between the transmitter and the receiver, the electromagnetic wave will bounce off objects in the environment and arrive at the receiver from different paths at different times. This cause multipath fading, which again can be roughly modeled as a power law function of the distance between the transmitter and receiver. Regardless of the model used (direct line-of-sight or multipath fading), the received power decreases as the distance between the transmitter and receiver increases [67].

For the tests carried out in this thesis, both the free space model and the multipath fading model were used, depending on the distance between the transmitter and receiver, as defined by the channel propagation model in [67] [68]. If the distance between the transmitter and receiver is less than a certain cross-over distance ($d_{crossover}$), the Friss free space model is used (d^2 attenuation), and if the distance is greater than $d_{crossover}$, the two-ray ground propagation model is used (d^4 attenuation). The cross-over point is defined as follows:

$$d_{crossover} = \frac{4\pi\sqrt{L}h_r h_t}{\lambda} \tag{5-1}$$

Where

$L \geq 1$ is the system loss factor not related to propagation,

h_r is the height of the receiving antenna above ground,

h_t is the height of the transmitting antenna above ground, and

λ is the wavelength of the carrier signal.

Based on the definition of $d_{crossover}$ as in Equation 5-1, if the distance between transmitter and receiver is less than $d_{crossover}$, the transmitted power is attenuated according to the Friss free space equation below:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi d)^2 L} \tag{5-2}$$

Where

$P_r(d)$ is the receive power given a transmitter-receiver separation of d,

P_t is the transmission power,

G_t is the gain of the transmitting antenna,

G_r is the gain of the receiving antenna,

λ is the wavelength of the carrier signal,

d is the distance between the transmitter and the receiver, and

$L \geq 1$ is the system loss factor not related to propagation.

Equation 5-2 models the attenuation when the transmitter and receiver have direct, line-of-sight communication, which will only occur if the transmitter and receiver are close to each other (i.e., $d < d_{crossover}$). If their distance is greater than $d_{crossover}$, the transmitted power is attenuated according to the two-ray ground propagation equation as follows:

$$P_r(d) = \frac{P_t G_t G_r {h_t}^2 {h_r}^2}{d^4} \qquad (5\text{-}3)$$

Where

$P_r(d)$ is the receive power given a transmitter-receiver separation of d,

P_t is the transmission power,

G_t is the gain of the transmitting antenna,

G_r is the gain of the receiving antenna,

h_r is the height of the receiving antenna above ground,

h_t is the height of the transmitting antenna above ground, and

d is the distance between the transmitter and the receiver.

In this case, the received signal comes from both the direct path and a ground-reflection path. Due to destructive interference when there is more than one path through which the signal arrives, the signal is attenuated as d^4 [67].

In the simulations described in this thesis, an omnidirectional antenna is used with the following parameters:

- $G_t = G_r = 1$
- $h_t = h_r = 1.5m$,

- no system loss $(L = 1)$,
- $914\ MHz$ radios, and
- $\lambda = 0.328\ m.$

With these parameter values, the cross-over distance , $d_{crossover}$ becomes $86.2\ m$ and Equations 5-2 and 5-3 are simplified to:

$$P_r = \begin{cases} 6.82 * 10^{-4}\frac{P_t}{d^2}: & d < 86.2\ m \\ 2.25\frac{P_t}{d^4}: & d \geq 86.2\ m \end{cases} \quad (5\text{-}4)$$

5.1.2. Radio Energy Dissipation Model

Currently, there has been a significant amount of research in the area of low-energy radios. Different assumptions about the radio characteristics, including energy dissipation in the transmit and receive models, significantly affect comparison result between different protocols. In this work, we assume a simple model where the transmitter dissipates energy to run the radio electronics and the power amplifier and the receiver dissipates energy to run the radio electronics [69].

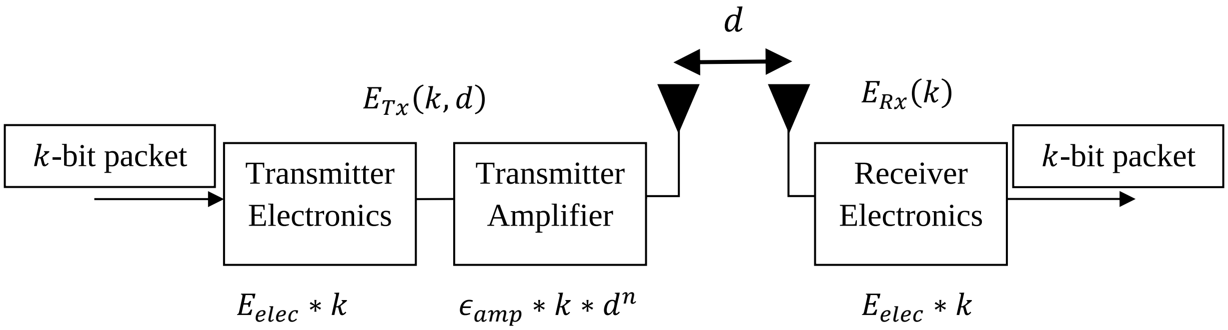


Figure 5-1: First order radio model (Radio energy dissipation model) [54]

The power attenuation is dependent on the distance between the transmitter and receiver. For relatively short distances, the propagation loss can be modeled as inversely proportional to d^2, whereas for longer distances, the propagation loss can be modeled as inversely proportional to d^4. Power control can be used to avert this loss by setting the power amplifier to ensure a certain power at the receiver. Thus, to transmit a k-bit message a distance d, the radio expends:

$$E_{Tx}(k,d) = E_{Tx-elec}(k) + E_{Tx-amp}(k,d) \tag{5-5}$$

$$E_{Tx}(k,d) = \begin{cases} E_{elec} * k + \epsilon_{friss-amp} * k * d^2, & d < d_{crossover} \\ E_{elec} * k + \epsilon_{two-ray-amp} * k * d^4, & d \geq d_{crossover} \end{cases} \tag{5-6}$$

Where E_{elec} is the energy dissipated per bit at the electronics for processing, $\epsilon_{friss-amp}$ is the energy consumed by the amplifier to transmit a bit at a distance shorter than $\mathrm{d_{crossover}}$, $\mathrm{\epsilon_{two-ray-amp}}$ for a longer distance and $\mathrm{d_{crossover}}$ is taken to be 87m for this test.

And to receive this message, the radio consumes:

$$E_{Rx}(k) = E_{Rx-elec}(k)$$

$$E_{Rx}(k) = E_{elec} * k \tag{5-7}$$

The electronics energy, E_{elec} depends on factors such as the digital coding, modulation, and filtering of the signal before it is sent to the transmitter amplifier [70]. For the simulation described in this thesis, the energy dissipated per bit in the transceiver electronics taken to be:

$$E_{elec} = 50 \text{ nJ/bit} \tag{5-8}$$

For 1 $Mbps$ transceiver, this means the radio electronics dissipates 50 mW when it is in operation (either transmitter or receiver).

The parameters $\epsilon_{friss-amp}$ and $\epsilon_{two-ray-amp}$ will depend on the required receiver sensitivity and the receiver noise figure, as the transmitter power needs to be adjusted so that the power at the receiver is above a certain threshold, $P_{r-thresh}$. We can work backwards from this receiver power threshold to determine the minimum transmitter power. If the radio bitrate is R_b, the transmission power, P_t is equal to the transmission energy per bit $E_{Tx-amp}(1,d)$ times the bitrate:

$$P_t = E_{Tx-amp}(1,d) * R_b \tag{5-9}$$

Inserting in the value of $E_{Tx-amp}(1,d)$ from Equation 5-6 gives:

$$P_t = \begin{cases} \epsilon_{friss-amp} * R_b * d^2 & : d < d_{crossover} \\ \epsilon_{two-ray-amp} * R_b * d^4 & : d \geq d_{crossover} \end{cases} \tag{5-10}$$

Using the channel models described in the previous section, the received power is:

$$P_r = \begin{cases} \dfrac{\epsilon_{friss-amp} R_b G_t G_r \lambda^2}{(4\pi)^2} & : d < d_{crossover} \\ \epsilon_{two-ray-amp} R_b G_t G_r {h_t}^2 {h_r}^2 & : d \geq d_{crossover} \end{cases} \tag{5-11}$$

The parameters $\epsilon_{friss-amp}$ and $\epsilon_{two-ray-amp}$ can be determined by setting Equation 5-11 equal to $P_{r-thresh}$:

$$\epsilon_{friss-amp} = \frac{P_{r-thresh}(4\pi)^2}{R_b G_t G_r \lambda^2} \tag{5-12}$$

$$\epsilon_{two-ray-amp} = \frac{P_{r-thresh}}{R_b G_t G_r {h_t}^2 {h_r}^2} \tag{5-13}$$

Therefore, the required transmission power, P_t, as a function of the receiver threshold and the distance between the transmitter and receiver is:

$$P_t = \begin{cases} \alpha_1 P_{r-thresh} d^{\,2} & : d < d_{crossover} \\ \alpha_2 P_{r-thresh} d^{\,4} & : d \geq d_{crossover} \end{cases} \tag{5-14}$$

Where $\alpha_1 = (4\pi)^2/(G_t G_r \lambda^2)$ and $\alpha_2 = 1/(G_t G_r {h_t}^2 {h_r}^2)$.

The receiver threshold $P_{r-thresh}$ can be determined using estimates for the noise at the receiver. If the thermal noise floor is $99\ dBm$ and the receiver noise figure is $17\ dB$, and we require a signal-to-noise ratio (SNR) of at least $30\ dB$ to receive the signal with no errors [71] [67], the minimum receive power $P_{r-thresh}$ for successful reception is

$$P_{r-thresh} \geq 30 + (-82) = -52\ dBm \tag{5-15}$$

Therefore, the received power must be at least $-52\ dBm$ or $6.3\ nW$ for successful reception of the packet. Inserting the values that will be used in the simulations ($G_t = G_r = 1, h_t = h_r = 1.5m, \lambda = 0.328\ m,$ and $R_b = 1\ Mbps$ into Equations 5-12 and 5-13 gives:

$$\epsilon_{friss-amp} = 10\ pJ/bit/m^2 \tag{5-16}$$

$$\epsilon_{two-ray-amp} = 0.0013\ pJ/bit/m^4 \tag{5-17}$$

These are the radio energy parameter that will be used in this work.

5.1.3. Beamforming Energy Model

In sensor networks, it is not the actual data itself that is important; rather, the processed data, which allows an end-user to determine something about the environment being monitored. For example, if the sensors are monitoring an area for surveillance purposes, the end-user does not need to see the data from all the individual sensors, but needs to know whether or not there has been an intrusion in the area being monitored. Therefore, automated methods of combining or aggregating the data into a small set of meaningful information are required [72] [73] [74] [75]. In addition to avoiding information overload, data aggregation (data fusion) can combine several unreliable data measurements to produce a more accurate signal.

One method of aggregating data is called beamforming [76] [77]. Beamforming combines signals from multiple sensors as follows:

$$y[n] = \sum_{i=1}^{N} \sum_{l=1}^{L} W_i[l] S_i[n-l] \tag{5-18}$$

Where $s_i[n]$ is the signal from the i^{th} sensor, $w_i[n]$ is the weighting filter for the i^{th} signal, N is the total number of sensors whose signals are being beamformed, and L is the number of taps in the filter. This algorithm is shown in Figure 5-2. The weighting filters are chosen to satisfy an optimization criterion, such as minimizing mean squared error (MSE) or maximizing signal-to-noise ratio (SNR). Various algorithms, such as least mean squared (LMS) error approach and the maximum power beamforming algorithm [77], has been developed to determine good weighting filters. These algorithms have various energy and quality tradeoffs [78]. For example, the LMS beamforming algorithm requires much less energy than the Maximum Power beamforming algorithm. In addition, the energy for LMS beamforming scales linearly with the number of sensors, while the energy for Maximum Power beamforming scales quadratically with the number of sensors. Therefore, the LMS beamforming algorithm is better-suited for implementation on a low-power microsensor node. Implementing the LMS beamforming algorithm on StrongARM-1100 (SA-1100) processer requires $5\ nJ/bit/signal$. Therefore, computation energy for beamforming E_{BF} is set to $5\ nJ/bit/signal$ [78] [79].

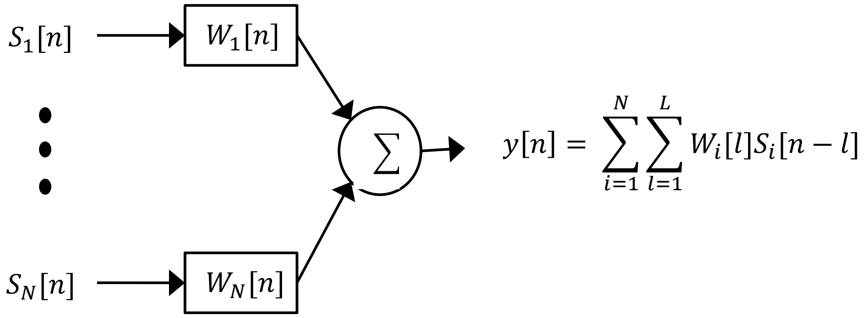


Figure 5-2: Block diagram of the beamforming algorithm. The individual sensor signals, $S_i[n]$ are filtered with weighting filters, $W_i[n]$ to get the beamformed signal, $y[n]$ [60].

5.2. Simulation Set-up

Matlab script has been written to model the various components and behaviors of the WSN under test. The simulations are run using MATLAB to evaluate the performance of the different protocols discussed in Chapter 4. In the simulations, a random 100-node network, illustrated in

Figure 5-3, is considered. The base station is placed at the center of the network, at location (x=50, y=50). The parameters that are used in the simulations are summarized in Tables 5-1 and 5-2.

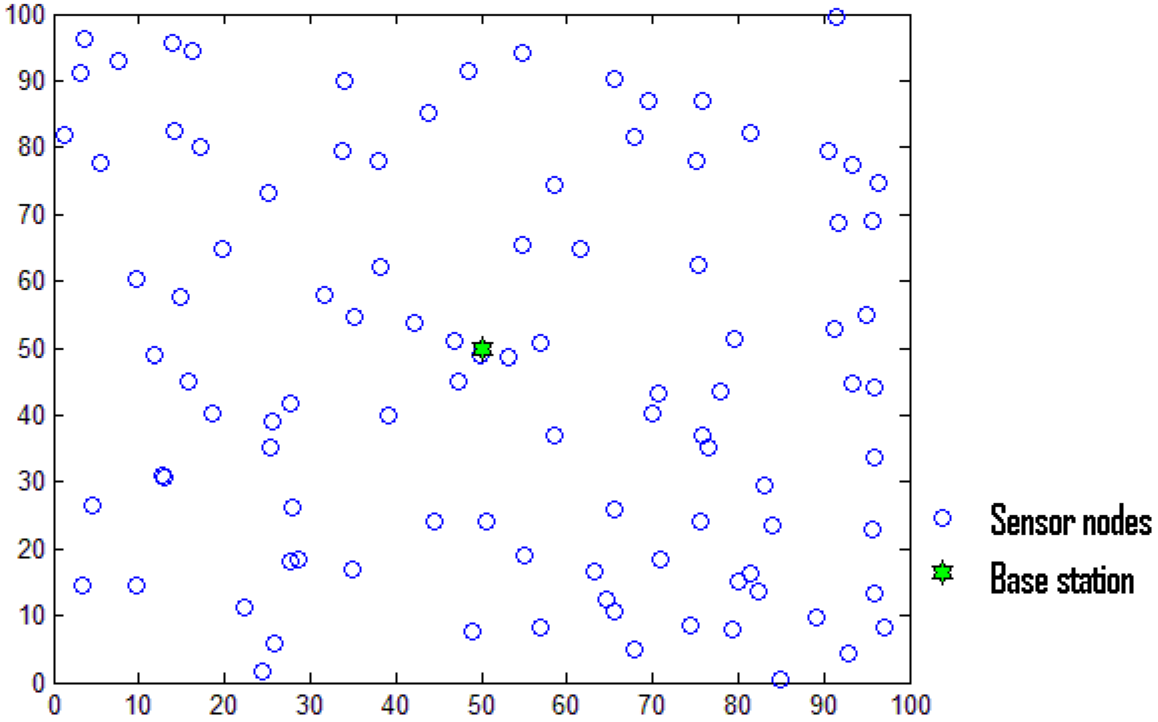


Figure 5-3: 100-node random test network. The base station is located at the center of the network, at location (x=50, y=50).

Table 5-1: Radio characteristics and parameter values [54]

Description	Parameter	Value
Minimum receiver power needed for successful reception	$P_{r-thresh}$	$6.3\ nW$
Radio electronics energy	E_{elec}	$50\ nJ/bit$
Compute energy for beamforming	E_{BF}	$5\ nJ/bit$

Bitrate	R_b	$1\ Mbps$
Antenna gain factor	G_t, G_r	1
Antenna height above the ground	h_t, h_r	$1.5\ m$
Signal frequency	f	$914MHz$
Cross-over distance for Friss and two-ray ground attenuation models	$d_{crossover}$	$87\ m$
Radio amplifier energy	$\epsilon_{friss-amp}$ $\epsilon_{two-ray-amp}$	$10\ pJ/bit/m^2$ $0.0013\ pJ/bit/m^4$

Table 5-2: Characteristics of the test network

Number of nodes	100
Network size	$100m\ x\ 100m$
Base station location	(50,50)
Packet size	$2000\ bits$
Initial node energy	$0.5\ J$

5.3. Simulation Results

For the simulations described in this section, Direct Transmission, LEACH, GAICH, and GCH are implemented. The metrics that are used to compare the proposed protocols; GAICH and GCH with LEACH and Direct Transmission are:

- **Network lifetime:** Sensor networks should function as long as possible. Different approaches can be used to measure network lifetime. One approach is to measure the time until the first node dies. Application-specific parameters, such as the time until the sensor network is no longer providing acceptable quality results (e.g., there are too many missed events) can also be used. In this test, the time at which the first node dies is taken as the lifetime of the network.

- **Total packets received at the base station:** Although quality is an application-specific and data-dependent quantity, one application independent method of determining quality is to measure the amount of data (number of actual data signals or number of data signals represented by an aggregate signal) received at the base station. The more data the base station receives, the more accurate its view of the remote environment will be. As a result the metric, total packets received at the base station, is taken as an important performance metric to compare the protocols discussed.
- **Amount of energy consumed per successful data report:** In addition to total packets received at the base station and system lifetime, amount of energy consumed per successful data report is an important metric that shows the efficient utilization of energy in the network by the routing protocols. However, the metrics such as delay of delivering data packets to the base station and overhead of running the algorithms are not taken into consideration.

5.3.1. Fixed Base Station Location

For the first set of simulations, each node begins with only 0.5 J of energy and an unlimited amount of data to send to the base station. Since the nodes have limited energy supply, they use up this energy during the course of simulation. When a node exhausts its energy, it is considered dead and can no longer transmit or receive data. For these simulations, energy is removed whenever a node transmits or receives data and whenever it performs data aggregation.

In Figure 5-4, the number of packets received at the base station in each round is plotted for each of the four protocols. GCH sends more packets in almost all of the rounds than LEACH. The number of packets sent in each round decreases sharply with increasing round number, once it starts decreasing form the maximum value in LEACH as compared to GAICH. This means the energy distribution of the network will be uniform in GAICH than LEACH. In general, from this figure, we can see that GCH and GAICH outperform LEACH with respect to the number of packets received at the base station per round.

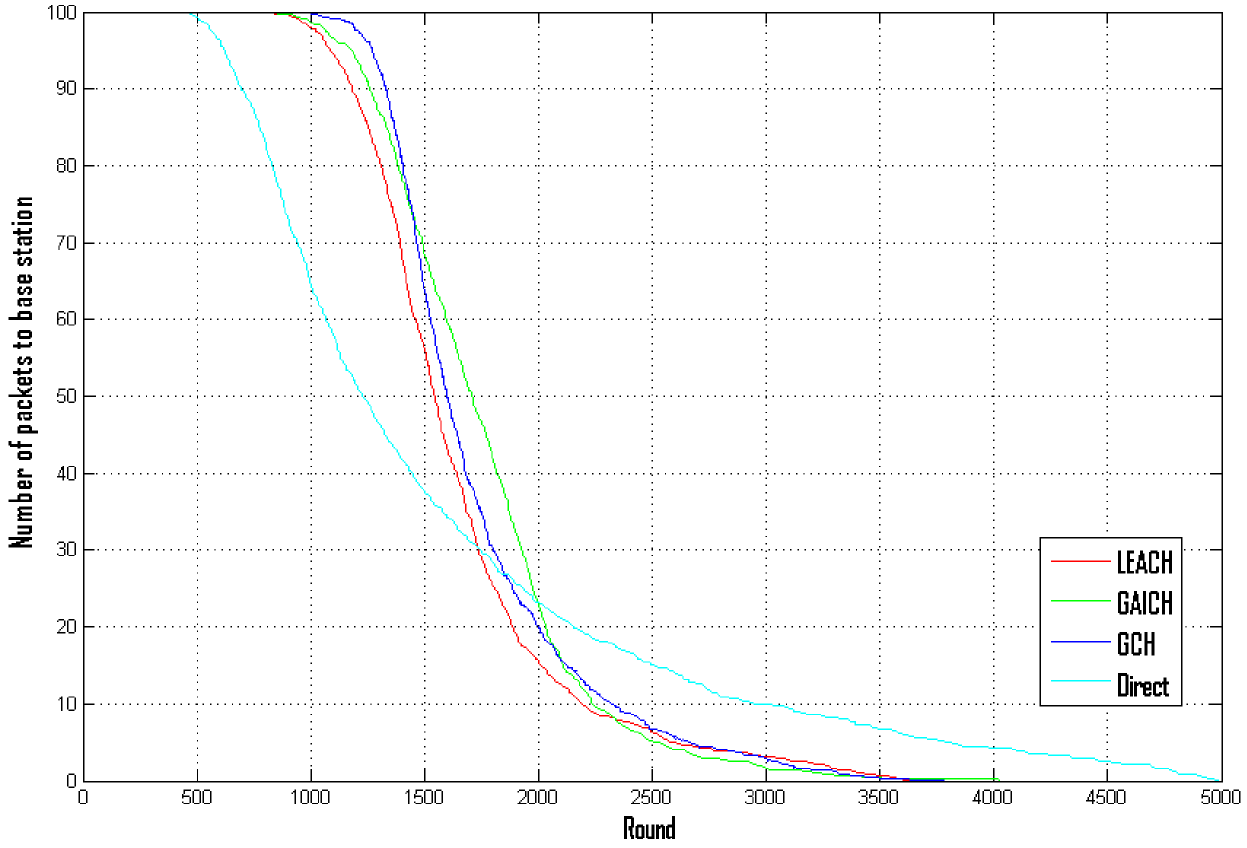


Figure 5-4: Number of packets sent to base station per round

The total packets sent to the base station over number of rounds are more in GCH and GAICH than in LEACH. This is clearly shown in Figure 5-5, the total packets received at the base station over number of rounds. This graph shows that GCH improved LEACH by 5% in the total packets sent to the base station, whereas GAICH improved LEACH by 6%.

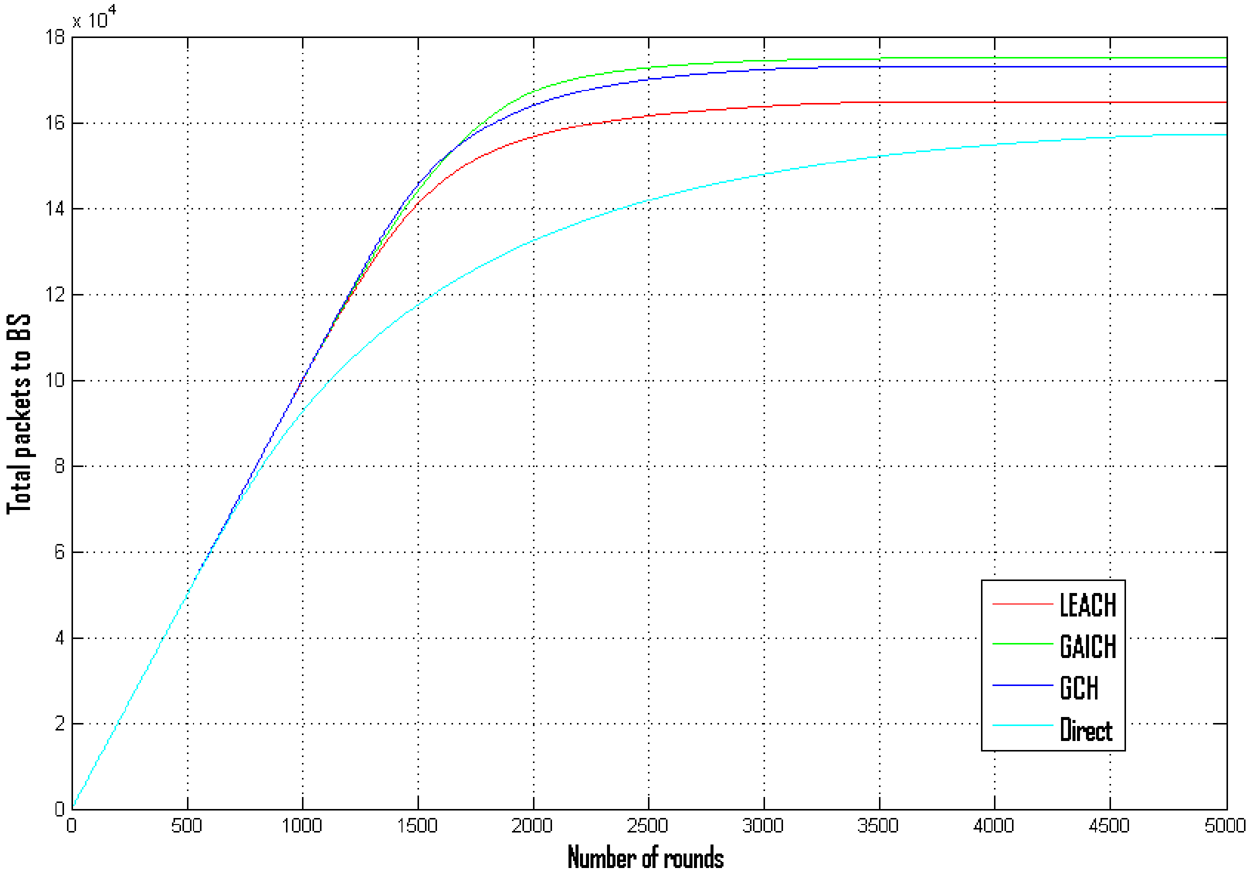


Figure 5-5: The total packets received at the base station over number of rounds.

As Direct Transmission does not involve data aggregation, all the packets that are sent to the base station are raw data. In Figure 5-6, we can see that the packets without aggregation received at the base station over number of rounds is more in Direct Transmission than LEACH, GCH, and GAICH. Since GAICH uses more cluster heads than LEACH, the packets without aggregation received at the base station in GAICH are less than in LEACH. After about 1200 rounds GCH takes a second level gridding which is four cluster heads per round. As a result the packets sent to the base station without aggregation starts increasing. Then, GCH takes the whole network as one grid in third level gridding; it becomes effectively Direct Transmission. Consequently, after about 1700 rounds, the packets without aggregation sent in GCH are more than LEACH and GAICH as shown in the figure. Generally, from Figure 5-6, we can conclude that more data aggregation is done in GAICH and GCH; as a result more energy is saved by these protocols.

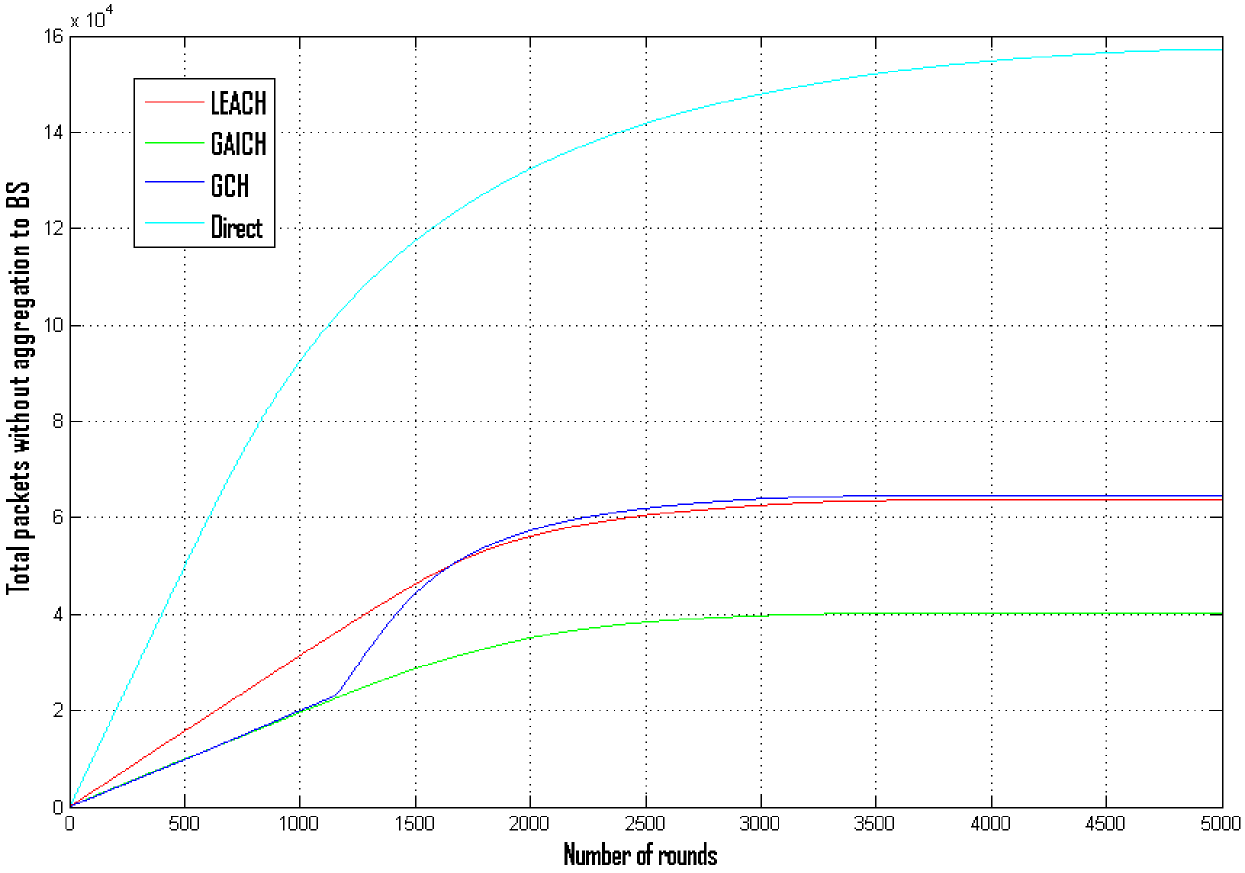


Figure 5-6: The total packets sent to the base station without aggregation over number of rounds.

The lifetimes of the different protocols are presented in Table 5-3. Here, GCH extends the time of the first node dies as compared to LEACH and GAICH by about 16%.

Table 5-3: Lifetimes of the different protocols where the base station is at the center of the network

Protocol	Round first node dies	Round last node dies
LEACH	941	3427
GAICH	946	3517

GCH	1100	3379
Direct Transmission	516	4821

Figure 5-7 shows the total number of dead nodes over number of rounds. While nodes remain alive for a long time in Direct Transmission, a much smaller amount of data has been transmitted to the base station. If the total number of nodes that remain alive per amount of data received at the base station is plotted, as shown in Figure 5-8, we see that nodes in GAICH and GCH can deliver more data than LEACH and Direct Transmission for the same number of dead node. Therefore, nodes in GAICH and GCH are better in using the available energy.

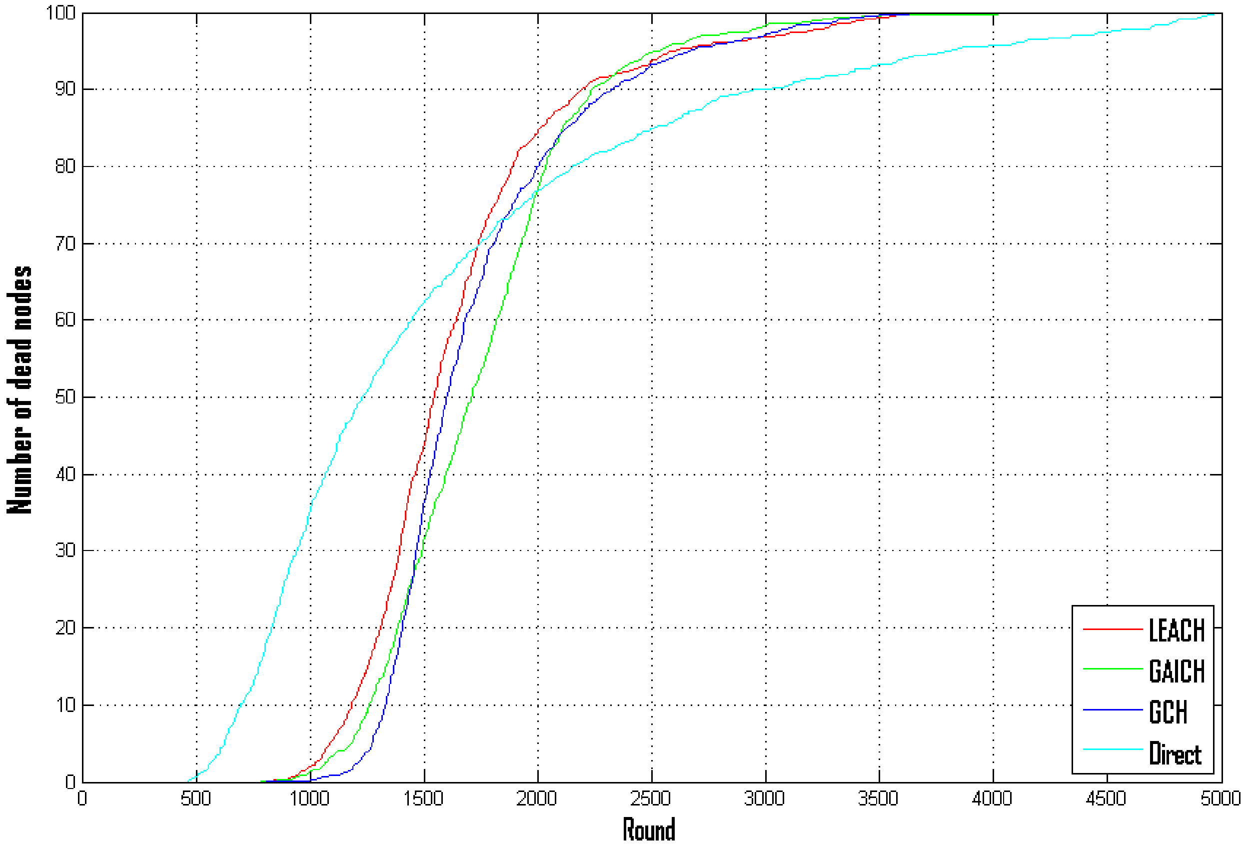


Figure 5-7: Number of dead nodes per round.

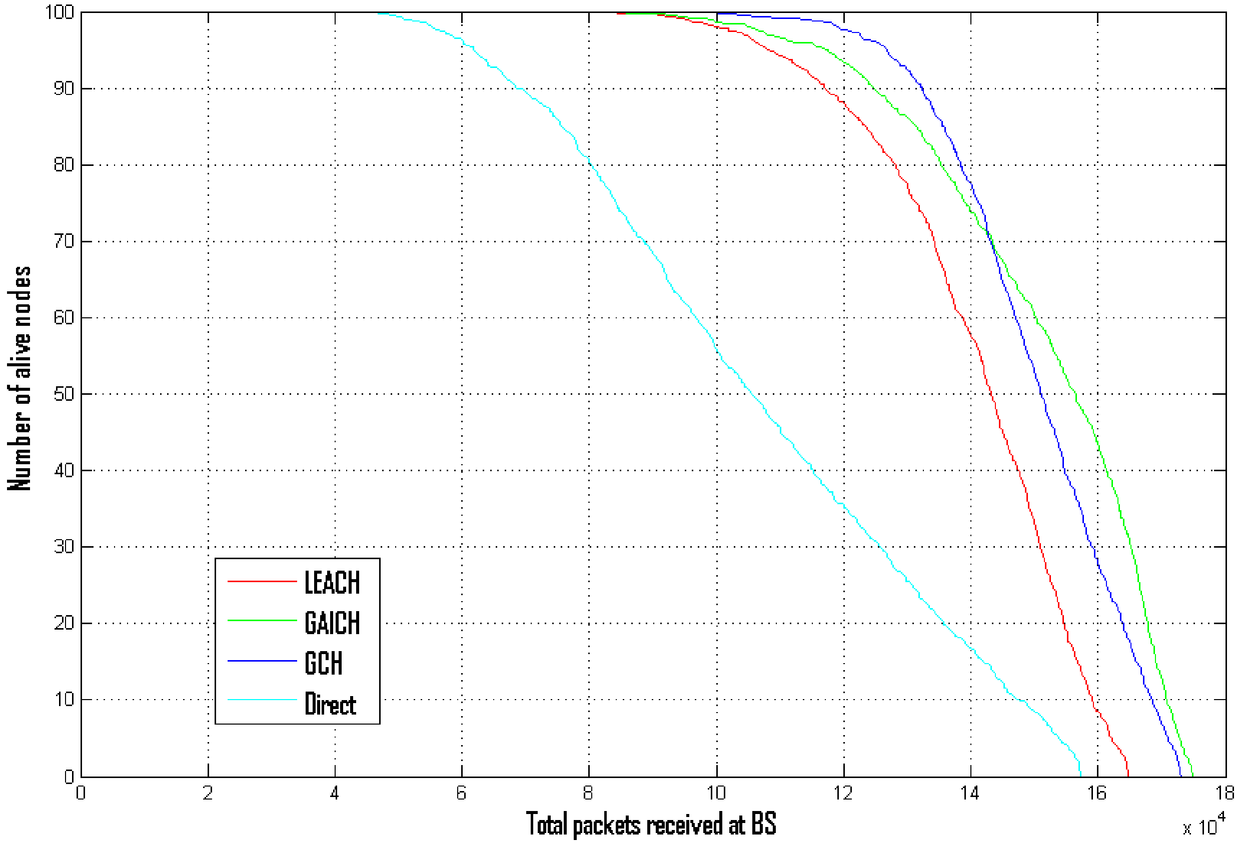


Figure 5-8: Number of nodes alive per total packets sent to the base station.

Figure 5-9 shows the total energy dissipated over number of rounds. For the same amount of total energy dissipated, GAICH and GCH run more rounds than LEACH and Direct Transmission. But at the end of the network lifetime, Direct Transmission run more rounds than the other protocols, because the nodes that remain alive are nodes nearer to the base station.

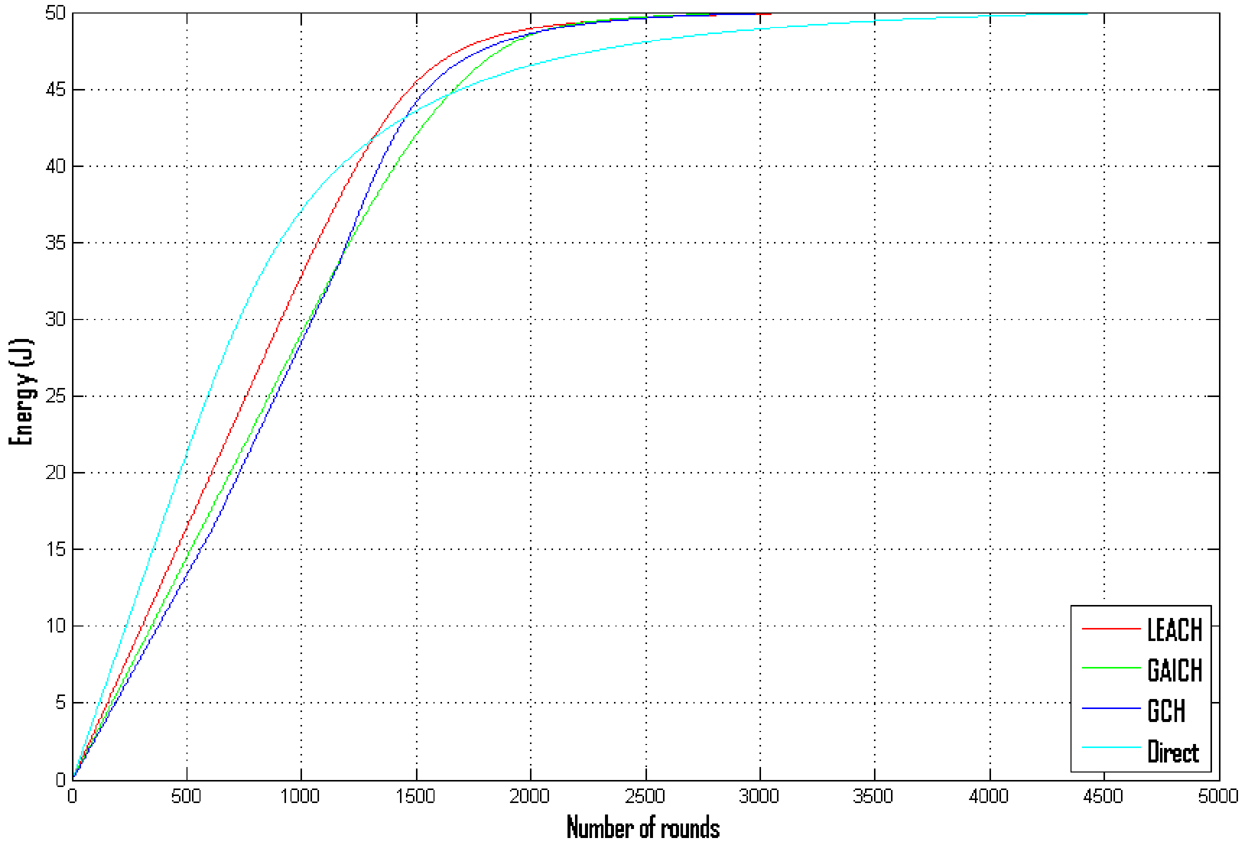


Figure 5-9: Total amount of energy dissipated in the system.

5.3.2. Varying the Base Station Location

The tests carried out in the previous section were for fixed base station location. In this section, tests are carried out for different locations of the base station with respect to the network shown in Figure 5-3.

Table 5-4 summarizes the performance comparisons between the protocols as the location of the base station varies from (x=50, y=50) to (x=50, y=150). From this table, It is seen that when the base station is in the center of the network (x=50, y=50), GAICH delivers 6% more packets to the base station as LEACH, whereas GCH delivers 5% more than LEACH. As the base station moves further away from the network, the performance of GAICH improves compared to

LEACH. For all base station locations simulated, GAICH performs better than LEACH by at least 4% and as much as 7%.

Table 5-4: Performance of the protocols as the base station location is varied.

Base Station Location from Network Center	Protocol	First Dead at Round	Last Dead at Round	Total Packets to BS	Performance Improvement over Direct Transmission (%)	Performance Improvement over LEACH (%)
(x=50 , y=50) 0 m	**LEACH**	941	3427	164717	5	0
	GAICH	946	3517	174939	11	6
	GCH	1100	3379	173041	10	5
	Direct T.	516	4821	157206	0	-5
(x=50 , y=100) 50 m	**LEACH**	697	3379	139698	34	0
	GAICH	575	3385	149893	44	7
	GCH	788	3166	143153	38	2
	Direct T.	211	4594	104059	0	-26
(x=50 , y=150) 100 m	**LEACH**	630	1175	96023	257	0
	GAICH	380	1615	100018	271	4
	GCH	383	1267	93815	248	-2
	Direct T.	105	736	26923	0	-72

Conclusions and Recommendations

6.1. Conclusions

Routing in sensor networks has attracted a lot of attention in the recent years and introduced new challenges compared to traditional data routing in wired networks. In this thesis, two hierarchical routing protocols: GAICH and GCH, are proposed with the objective of reducing power consumption in the network. GAICH is a genetic algorithm based routing protocol that minimizes global energy usage by distributing the load to all the nodes at different points in time. GAICH takes into consideration different parameters that affect the energy dissipation directly or indirectly. The second protocol proposed, GCH, is a clustering based routing protocol that divides the network into virtual grids based on the current average energy of the network. As GCH uses round-robin scheduling of the cluster head role in each grid, it distributes the overall energy dissipation of the nodes uniformly throughout the network.

GAICH outperforms LEACH with respect to network lifetime, total packets received at the base station and amount of energy consumed per successful data report, because it selects cluster heads that result in optimum energy consumption. It defines the fitness function for the genetic algorithm by taking parameters such as distances to base station, density, centrality and residual energy of cluster heads. These parameters affect the energy consumption and network life time of the WSN directly or indirectly. At different times, each node has the burden of acquiring data from the nodes in the cluster, fusing the data to obtain an aggregate signal, and transmitting this aggregate signal to the base station. This protocol produces a better cluster head distribution than LEACH, as it has global knowledge of the location of all nodes in the network and takes into consideration different parameters that affect the overall energy distribution of the network. However, this requires that nodes be equipped with GPS or other location-finding algorithms. In addition, the computation of the genetic algorithm causes latency in the network.

GCH organizes the network into variable number of virtual girds based on the current average energy of the network. Cluster heads are selected from nodes in each grid in a round-robin fashion. This makes the load of receiving, aggregating and transmitting data of nodes in the cluster to be distributed in every corner of the network. It is centralized and requires knowledge of the global network. GCH outperforms LEACH in terms of the total data received at the base station and the system lifetime of the network. Even if the performance of GCH is better, the idea of dynamic gridding and round-robin cluster head rotation brings extra overhead, which may diminish the gain in energy consumption. Moreover, in the simulation GCH performs less than GAICH. This is due to the fact that the gridding levels taken are only three. To increase, the performance of GCH, it should be run for different number of gridding levels depending on the specific sensor network under consideration.

Distributing the energy among the nodes in the network is effective in reducing energy dissipation from a global perspective and enhancing system lifetime. Specifically for the base station location at the center of network, the simulations show that:

- GCH improved LEACH by 5% in the total packets sent to the base station, whereas GAICH improved LEACH by 6%.
- The time of the first node death in GCH occurs over 16% later than its death in LEACH.
- For the same amount of total energy dissipated, GAICH and GCH run more rounds than LEACH.
- Nodes in GAICH and GCH can deliver more data than LEACH for the same number of dead nodes.

6.2. Recommendations

The simulations of the proposed protocols: GAICH and GCH, have short comings in performance modeling. Issues such as delay of delivering data packets to the base station and overhead of running the algorithms are not taken into consideration. More convincing conclusions could be obtained if these important issues were considered in the performance model.

Although MATLAB is widely used in simulations of WSN routing protocols, more realistic simulations can be carried out with NS2, which incorporates different built in MAC protocols for WSNs. As a result, the overheads associated with implementing GAICH and GCH can be studied thoroughly. The study of these protocols using NS2 and other WSN specific simulators such as Castalia, OMNeT++ and J-Sim can be carried out to investigate the overall benefits obtained. While these simulators are used in the WSN simulations, there are drawbacks if our aim is in configuring practical nodes. So, in addition to these simulators, a study by using hardware dependent simulation tools such as TOSSIM, ATEMU and Avrora can be done.

GAICH improved the lifetime of the WSN effectively using genetic algorithm. Further investigations can be done in using other intelligent algorithms instead of GA. Nature-inspired optimization algorithms such as ant colony, particle swarm, bacterial foraging and simulated annealing optimization can be used in optimizing the energy distribution in the WSN. Moreover, hybrid solutions can be considered by combining these algorithms in choosing the cluster head nodes and selecting the cluster members.

Further investigation into the mobility of sensor nodes should be done to see the wider applications of the proposed protocols. It is also expected that these protocols can offer significant improvement on performance and energy efficiency of mobile sensor networks if further research on mobility models is carried out. Moreover, the work can be extended by including clustering of clusters or multi-hop routing between cluster heads.

References

[1] C. Y. Chong and S. P. Kumar, *''Sensor Networks: Evolution, Opportunities and Challenges,''* Proceedings of the IEEE, Vol. 91, No. 8, August 2003, pp. 1247-1254.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, *"A survey on Sensor Networks,"* IEEE Communications Magazine, vol. 40, Issue: 8, pp. 102-114, August 2002.

[3] W. Su et al., *''Communication Protocols for Sensor Networks,''* in Wireless Sensor Networks, Kluwer Academic, New York, 2004.

[4] Kazem Sohraby, Daniel Minoli, and Taieb Znati, *"Wireless Sensor Networks: Technology, Protocols, and Applications,"* John Wiley & Sons, Inc, 2007.

[5] G. Simon, M. Maroti, A. Ledeczi, G. Balogh, B. Kusy, A. Nadas, G. Pap, J. Sallai, K. Frampton, *"Sensor network-based counter sniper system,"* in the Proceedings of the Second International Conference on Embedded Networked Sensor Systems (Sensys), Baltimore, MD, 2004.

[6] J. Yick, B. Mukherjee, D. Ghosal, *"Analysis of a Prediction-based Mobility Adaptive Tracking Algorithm,"* in the Proceedings of the IEEE Second International Conference on Broadband Networks (BROADNETS), Boston, 2005.

[7] T. Gao, D. Greenspan, M. Welsh, R.R. Juang, A. Alm, *"Vital signs monitoring and patient tracking over a wireless network,"* in the Proceedings of the 27th IEEE EMBS Annual International Conference, 2005.

[8] M. Castillo Effen, D.H. Quintela, R. Jordan, W. Westhoff, W. Moreno, *"Wireless sensor networks for flash-flood alerting,"* in the Proceedings of the Fifth IEEE International Caracas Conference on Devices, Circuits, and Systems, Dominican Republic, 2004.

[9] K. Lorincz, D. Malan, T.R.F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, S. Moulton, *"Sensor networks for emergency response: challenges*

and opportunities," Pervasive Computing for First Response (Special Issue), IEEE Pervasive Computing, October–December 2004.

[10] G. Wener-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, M. Walsh, *"Deploying a wireless sensor network on an active volcano,"* Data-Driven Applications in Sensor Networks (Special Issue), IEEE Internet Computing, March/April 2006.

[11] C. Perkins, "*Ad Hoc Networks,"* Addison-Wesley, Reading, MA, 2000.

[12] C. Shen, C. Srisathapornphat, and C. Jaikaeo, *"Sensor Information Networking Architecture and Applications,"* IEEE Pers. Communication, Aug. 2001, pp. 52–59.

[13] G. Hoblos, M. Staroswiecki, and A. Aitouche, *"Optimal Design of Fault Tolerant Sensor Networks,"* IEEE International Conf. Cont. Apps., Anchorage, AK, Sept. 2000, pp. 467–72.

[14] N. Bulusu et al., *"Scalable coordination for wireless sensor networks: Self-configuring localization systems,"* In Proceddings of the 6th International Symposium on Communication Theory and Applications (ISCTA'01), Ambleside, UK, July 2001.

[15] J. M. Rabaey et al., *"PicoRadio Supports Ad Hoc Ultra- Low Power Wireless Networking,"* IEEE Comp. Mag., 2000, pp. 42–48.

[16] G. J. Pottie and W. J. Kaiser, *"Wireless Integrated Network Sensors,"* Communication ACM, vol. 43, no. 5, May 2000, pp. 551-58.

[17] J. M. Kahn, R. H. Katz, and K. S. J. Pister, *"Next Century Challenges: Mobile Networking for Smart Dust,"* Proc. ACM MobiCom '99, Washington, DC, 1999, pp. 271–78.

[18] E. Shih et al., *"Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks,"* Proc. ACM MobiCom '01, Rome, Italy, July 2001, pp. 272–86.

[19] C. Intanagonwiwat, R. Govindan and D. Estrin, *"Directed diffusion: A scalable and robust communication paradigm for sensor networks"*, in the Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking, Boston, MA, August 2000.

[20] A. Woo, and D. Culler, *"A Transmission Control Scheme for Media Access in Sensor Networks,"* Proc. ACM MobiCom '01, Rome, Italy, July 2001, pp.221–35.

[21] H. Cam, S. Ozdemir, P. Nair, and D. Muthuavinashippan,*"ESPDA: Energy-Efficient and Secure Pattern Based Data Aggregation for Wireless Sensor Networks,"* in press, IEEE Sensor, Toronto, Canada, 2003.

[22] Perrig, A. et al., *"Security in wireless sensor networks,"* Communications of the ACM (CACM), Wireless sensor networks, Special Issue: Wireless sensor networks, vol. 47, issue: 6, pp. 53-57, June, 2004.

[23] Avancha, S. et al., *"Wireless Sensor Networks,"* Kluwer Academic/Springer Verlag Publishers, 2003.

[24] E. Shi, A. Perrig, *"Designing Secure Sensor Networks,"* IEEE Wireless Communications, pp.38- 43, December, 2004.

[25] B. Krishnamachari,*"On the Complexity of Distributed Self-Configuration in Wireless Networks,"* Kluwer Academic Springer Publishers, Telecommunication Systems, vol. 22, issue: 1-4, pp. 33-59, 2003

[26] Iyer, R. and L. Kleinrock, *"QoS Control for Sensor Networks,"* presented at the IEEE International Communications Conference (ICC' 03), Anchorage, AK, May 11-15. 2003.

[27] J. Kay, J. Frolik, *"Quality of Service Analysis and Control for Wireless Sensor Networks,"* In Proc. of the 21st International Conf. on Mobile Ad-Hoc and Sensor Systems (MASS'04), pp. 359-368, Fort Lauderdale, Florida, USA, 25-27 October, 2004.

[28] Eiko Yoneki, Jean Bacon,*"A survey of Wireless Sensor Network technologies: research trends and middleware's role,"* Technical Report, no: 646, UCAM–CL-TR-646, [Available from the World Wide Web http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR- 646.pdf]

[29] Kay Romer and Friedemann Mattern,*"The Design Space of Wireless Sensor Networks,"* IEEE Wireless Communications, pp. 54- 61, Dec. 2004. [Also, available: http://www.vs.inf.ethz.ch/publ/papers/wsn-designspace.pdf]

[30] F. Ye et al., *"A Two-tier Data Dissemination Model for Large-scale Wireless Sensor Networks,"* in the Proceedings of Mobicom'02, Atlanta, GA, Septmeber, 2002

[31] S. Tilak et al., *"A Taxonomy of Wireless Microsensor Network Models,"* in ACM Mobile Computing and Communications Review (MC2R), June 2002.

[32] Kemal Akkaya and Mohamed Younis, *"A Survey on Routing Protocols for Wireless Sensor Networks"* Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, 2003

[33] J.-H. Chang and L. Tassiulas, *"Maximum Lifetime Routing in Wireless Sensor Networks,"* in the Proceedings of the Advanced Telecommunications and Information Distribution Research Program (ATIRP'2000), College Park, MD, March 2000.

[34] K. Kalpakis, K. Dasgupta and P. Namjoshi, *"Maximum lifetime data gathering and aggregation in wireless sensor networks,"* in the Proceedings of IEEE International Conference on Networking (NETWORKS '02), Atlanta, GA, August 2002.

[35] K. Akkaya and M. Younis, *"An Energy-Aware QoS Routing Protocol for Wireless Sensor Networks,"* in the Proceedings of the IEEE Workshop on Mobile and Wireless Networks (MWN 2003), Providence, Rhode Island, May 2003.

[36] T. He et al., *"SPEED: A stateless protocol for real-time communication in sensor networks,"* in the Proceedings of International Conference on Distributed Computing Systems, Providence, RI, May 2003.

[37] W. Heinzelman, J. Kulik, and H. Balakrishnan, *“Adaptive protocols for information dissemination in wireless sensor networks,”* in the Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking , Seattle, WA, August 1999.

[38] D. Braginsky and D. Estrin, *"Rumor Routing Algorithm for Sensor Networks,"* in the Proceedings of the First Workshop on Sensor Networks and Applications (WSNA), Atlanta, GA, October 2002.

[39] C. Schurgers and M.B. Srivastava, *“Energy efficient routing in wireless sensor networks,”* in the MILCOM Proceedings on Communications for Network-Centric Operations: Creating the Information Force, McLean, VA, 2001.

[40] M. Chu, H. Haussecker, and F. Zhao, *"Scalable Information-Driven Sensor Querying and Routing for ad hoc Heterogeneous Sensor Networks,"* The International Journal of High Performance Computing Applications, Vol. 16, No. 3, August 2002.

[41] A. Manjeshwar and D. P. Agrawal, *“TEEN: A Protocol for Enhanced Efficiency in Wireless Sensor Networks,"* in the Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA, April 2001.

[42] Y. Yao and J. Gehrke, *“The cougar approach to in-network query processing in sensor networks,”* in SIGMOD Record, September 2002.

[43] R. Shah and J. Rabaey, *"Energy Aware Routing for Low Energy Ad Hoc Sensor Networks"*, in the Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Orlando, FL, March 2002.

[44] N. Sadagopan et al., *“The ACQUIRE mechanism for efficient querying in sensor networks,”* in the Proceedings of the First International Workshop on Sensor Network Protocol and Applications, Anchorage, Alaska, May 2003.

[45] S. Hedetniemi and A. Liestman, *“A survey of gossiping and broadcasting in communication networks,” Networks,* Vol. 18, No. 4, pp. 319-349, 1988.

[46] D. Ganesan, et al., *"Highly Resilient, Energy Efficient Multipath Routing in wireless Sensor Networks,"* in Mobile Computing and Communications Review, Vol. 1, No. 2, 2002

[47] V. Rodoplu and T.H. Ming, *"Minimum energy mobile wireless networks,"* IEEE Journal of Selected Areas in Communications, Vol. 17, No. 8, pp. 1333-1344, 1999.

[48] L. Li and J. Y Halpern, *“Minimum energy mobile wireless networks revisited,”* in the Proceedings of IEEE International Conference on Communications, Helsinki, Finland, June 2001

[49] Y. Xu, J. Heidemann, and D. Estrin, *"Geography-informed energy conservation for ad hoc routing,"* in the Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking, Rome, Italy, July 2001.

[50] Y. Yu, D. Estrin, and R. Govindan, *“Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks,”* UCLA Computer Science Department Technical Report, May 2001.

[51] B. Karp and H. T. Kung, *“GPSR: Greedy perimeter stateless routing for wireless sensor networks,”* in the Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking, Boston, MA, August 2000.

[52] A. Buczak and V. Jamalabad, *"Self-organization of a Heterogeneous Sensor Network by Genetic Algorithms,"* Intelligent Engineering Systems Through Artificial Neural Networks, C.H. Dagli, et. (eds.), Vol. 8, pp. 259-264, ASME Press, New York, 1998.

[53] C.R. Lin and M. Gerla, *"Adaptive Clustering for Mobile Wireless Networks,"* IEEE Journal on Selected areas in Communications, Vol. 15, No. 7, September 1997.

[54] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, *"Energy-efficient communication protocol for wireless sensor networks,"* in the Proceeding of the Hawaii International Conference on System Sciences, Hawaii, January 2000.

[55] S. Lindsey and C. S. Raghavendra, *"PEGASIS: Power Efficient GAthering in Sensor Information Systems,"* in the Proceedings of the IEEE Aerospace Conference, Big Sky, Montana, March 2002.

[56] S. Lindsey, C. S. Raghavendra and K. Sivalingam, *"Data Gathering in Sensor Networks using the Energy*Delay Metric,"* in the Proceedings of the IPDPS Workshop on Issues in Wireless Networks and Mobile Computing, San Francisco, CA, April 2001.

[57] A. Manjeshwar and D. P. Agrawal, *"APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks,"* in the Proceedings of the 2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile computing, Ft. Lauderdale, FL, April 2002.

[58] M. Younis, M. Youssef and K. Arisha, *"Energy-Aware Routing in Cluster-Based Sensor Networks"*, in the Proceedings of the 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS2002), Fort Worth, TX, October 2002.

[59] L. Subramanian and R. H. Katz, *"An Architecture for Building Self Configurable Systems,"* in the Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing, Boston, MA, August 2000.

[60] Wendi Beth Heinzelman, *"Application-Specific Protocol Architectures for Wireless Networks,"* PhD thesis , Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, June 2000

[61] John H. Holland, *"Adaptation in natural and artificial systems,"* The University of Michigan Press, Ann Harbor, MI, 1975

[62] *http://en.wikipedia.org/wiki/Genetic_algorithm*

[63] David E. Goldberg, *“Genetic algorithms in search, optimization and machine learning,”* Addison-Wesley, Reading, MA, 1985

[64] Sajid Hussain, Abdul Wasey Matin, Obidul Islam, *"Genetic Algorithm for Hierarchical Wireless Sensor Networks,"* Journal of Networks, Vol. 2, No. 5, September 2007

[65] Jamshid Shanbehzadeh, Saeed Mehrjoo, Abdolhossein Sarrafzadeh, *"An Intelligent Energy Efficient Clustering in Wireless Sensor Networks"* in the Proceedings of IMECS, Hong Kong, March, 20011,

[66] T. M. Mitchell, “*Machine Learning,”* McGraw Hill, 1997.

[67] T. Rappaport, *“Wireless Communications: Principles and Practice,”* Prentice-Hall, Inc., New Jersey, 1996.

[68] J. Broch, D. Maltz, D.Johnson, Y. Hu, and J. Jetcheva, *“A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols,”* In Proc. 4th ACM International Conference on Mobile Computing and Networking (Mobicom'98), October 1998

[69] D. Smithgall, *“Toward the 60 gm Wireless Phone,”* In Proceedings of the 1998 Radio and Wireless Conference (RAWCON' 98), pages 157-159, August 1998

[70] S. Sheng, L. Lynn, J. Peroulas, K. Stone, I. O'Donnell, and R. Brodersen, *“A Low Power CMOS Chipset for Spread Spectrum Communications,”* In Proceedings of the 42nd Solid-State Circuits Conference (ISSCC' 96), Feburary 1996

[71] P. Chandwick, *“Sensitivity of Range in WLAN Receivers,”* In IEE Colloquium on Radio LANs and MANs, 1995

[72] R. Brooks and S. Iyengar, *“Multi-Sensor Fusion,”* Prentice-Hall, Inc., Upper Saddle River, NJ, 1998.

[73] D. Hall, *“Mathematical Techniques in Multisensor Data Fusion,”* Artech House, Boston, MA, 1992.

[74] L. Klein, *“Sensor and Data Fusion Concepts and Applications,”* SPIE Optical Engr Press, WA, 1993.

[75] P. Varshney, *“Distributed Detection and Data Fusion,”* Springer-Verlag, New York, 1997.

[76] A. V. Oppenheim, *“Applications of Digital Signal Processing,”* Prentice-Hall, Inc., 1978

[77] K. Yao, R. Hudson, C. Reed, D. Chen, and F. Lorenzelli, *“Blind Beamforming on a Randomly Distributed Sensor Array System,”* in Proceedings of the 1998 IEEE Workshop on Signal Processing Systems (SiPS ’98), October 1998.

[78] A. Wang, W. Heinzelman, and A. Chandrakasan, *“Energy-Scalable Protocols for Battery-Operated Microsensor Networks,”* in Proceedings of the 1999 IEEE Workshop on Signal Processing Systems (SiPS ’99), pages 483-492, October 1999.

[79] J. Goodman, A. Chandrakasan, and A. Dancy, *“Design and Implementation of a Scalable Encryption Processor with Embedded Variable DC/DC Converter,”* In Proceedings of the 36th Design Automation Conference (DAC '99), pages 855-860, June 1999