An Energy Efficient Zone Based Data Gathering Approach in Wireless Sensor Networks

Abu Sayed Chowdhury

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

September 2012

An Energy Efficient Zone Based Data Gathering Approach in Wireless Sensor Networks

A Thesis Submitted to the
Department of Computer Science and Engineering
of
Bangladesh University of Engineering and Technology

in Partial Fulfillment of the Requirement
for the Degree of
MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

by

Abu Sayed Chowdhury

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
September 2012

The thesis entitled **"An Energy Efficient Zone Based Data Gathering Approach in Wireless Sensor Networks"** submitted by Abu Sayed Chowdhury, Student No.: 0409052008P, Session: April, 2009 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING on September 29, 2012.

BOARD OF EXAMINERS

1. ____

(Dr. Mahmuda Naznin) — **Chairman** (Supervisor)
Associate Professor
Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology,
Dhaka - 1000, Bangladesh.

2. ____

(Dr. Abu Sayed Md. Latiful Hoque) — **Member** (Ex-officio)
Professor and Head
Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology,
Dhaka - 1000, Bangladesh.

3. ____

(Dr. Md. Mostofa Akbar) — **Member**
Professor
Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology,
Dhaka - 1000, Bangladesh.

4. ____

(Dr. Md. Humayun Kabir) — **Member**
Professor
Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology,
Dhaka - 1000, Bangladesh.

5. ____

(Dr. Md. Rashedur Rahman) — **Member** (External)
Assistant Professor
Department of Electrical Engineering and Computer Science,
North South University, Dhaka - 1229, Bangladesh.

Declaration

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Signature of the candidate

____

(Abu Sayed Chowdhury)

Dedication

To my beloved parents

Contents

Declaration ii

Dedication iii

Acknowledgement xiii

Abstract xiv

1 Introduction 1

1.1 Overview 1

1.2 Wireless Sensor Networks 1

1.3 Challenges in Applications of WSNs 2

1.3.1 Applications 3

1.3.2 Challenges 5

1.4 Challenges in Data Gathering 6

1.4.1 Data Gathering 6

1.4.2 Challenges 7

1.5 Thesis Outlines 8

2 Related Work 9

2.1 Overview 9

2.2 Grid Based Approaches 9

2.2.1 Static Grid and Dynamic Grid Approaches 9

2.2.2 Dynamic Grid Switch Approach 11

2.2.3 Steiner Points Protocol 12

2.3 Tree Based Approaches 13

2.3.1 2-Approximation Double Tree Algorithm 13

2.3.2 Maximum Lifetime Data Gathering Tree (MLDGT) 14

2.4 Cluster Based Approaches 15
2.4.1 LEACH and Its Variants 15
2.4.2 Cluster-based Data Collection scheme with Direct Sink Access (CDC-DSA) 18
2.4.3 Novel Energy Efficient Cluster-based Algorithm 18
2.4.4 Tree Based Clustering (TBC) 20
2.4.5 Zone-based Fast Data Aggregation Tree (ZFDAT) 21
2.4.6 Cocentric Data Aggregation Model 21
2.4.7 Energy Balanced Data Gathering (EBDG) 22

3 Problem Domain **25**
3.1 Overview 25
3.2 System Model 25
3.2.1 Preliminaries 25
3.2.2 Assumptions and Challenges 27
3.3 Data Gathering Methodology 28
3.3.1 Zone Leader Selection 28
3.3.2 Handling Multiple Clusters in a Zone 28
3.4 Overview of ZDG 33
3.4.1 Details of ZDG 33
3.4.2 Data Gathering and Sending to Sink 34
3.4.3 Energy Efficiency and Deadlock Avoidance 36

4 Analysis of ZDG **38**
4.1 Overview 38
4.2 Energy Consumption by a Sensor 38
4.3 Energy Consumption Calculation for ZDG 40
4.3.1 Energy Consumption for Zone Formation 40
4.3.2 Energy Consumption for Sending Data 41
4.3.3 Energy Consumption for Inter-zone Communication 42
4.3.4 Energy Consumption for Intra-zone Communication 43
4.3.5 Energy Consumption for Data Aggregation 43
4.3.6 Energy Consumption in Data Communications per Round 44
4.3.7 Total Energy Consumption per Round in a Zone 45

4.4 Cost Analysis of ZDG 46

5 Experimental Results **47**

5.1 Overview 47

5.2 Simulation Environment 47

5.2.1 Simulation Platform 47

5.2.2 Sensor Node Architecture in OMNeT++ 48

5.3 Selected Methods and Types of Messages for Simulation 50

5.4 Simulating a Sensor Node 50

5.5 Simulation Metrics 50

5.6 Simulation Results 51

5.6.1 Building Scenario 51

5.6.2 Performance Evaluation 51

6 Conclusion and Future Work **57**

6.1 Conclusion 57

6.2 Future Work 58

Bibliography **59**

Appendix **64**

A Simulation Runs **64**

List of Figures

1.1 A typical sensor node structure 2
1.2 Some current commercial sensor nodes 2
1.3 A wireless sensor network for data collection in a crop field 3
1.4 A scenario of tracking the enemy using sensor networks 4

2.1 The notion of improvement 14
2.2 An example of cluster formation 15
2.3 Illustration of LEACH operation 16
2.4 Routing of Multihop-LEACH protocol 17
2.5 Illustration of network division in EBDG 22
2.6 The partition of coronas C_i and C_{i+1} in EBDG 23

3.1 A sample network divided in zones 26
3.2 Illustrating buffer limitation of the ZL in a zone 27
3.3 Illustrating calculation of hop count, M in a zone 29
3.4 Illustrating cluster formation when $M = 2$ 31
3.5 The details of zone based data gathering method ZDG 34
3.6 Scenario of data transmission between multiple clustered zones . . 35
3.7 A scenario of deadlock situation 36
3.8 Remedy of deadlock situation 37

4.1 Heinzelman's first order radio model 39
4.2 Current of boards in sensor node MICA2DOT (MPR 500) 40

5.1 A simulated sensor node structure 48
5.2 Comparison of the energy consumption with EBDG and Multihop-LEACH for $Buffer\_Cap = 17$ 52

5.3 Comparison of the network lifetime with EBDG and Multihop-LEACH for $Buffer\_Cap = 17$ 52
5.4 Comparison of the throughput with EBDG and Multihop-LEACH for $Buffer\_Cap = 17$ 53
5.5 Comparison of the throughput/energy consumption ratio with EBDG and Multihop-LEACH for $Buffer\_Cap = 17$ 53
5.6 Comparison of the energy consumption of ZDG for $Buffer\_Cap = 17$ and $Buffer\_Cap = 53$ 54
5.7 Comparison of the network lifetime of ZDG for $Buffer\_Cap = 17$ and $Buffer\_Cap = 53$ 54
5.8 Comparison of the throughput of ZDG for $Buffer\_Cap = 17$ and $Buffer\_Cap = 53$ 55
5.9 Energy consumption per zone of ZDG 55

List of Algorithms

3.1 Zone leader selection algorithm 28

3.2 Cluster formation algorithm 32

List of Tables

5.1 Simulation parameters 51

A.1 Experimental results of EBDG (Average) 64
A.2 Experimental results of Multihop-LEACH (Average) 65
A.3 Experimental results of ZDG for *Buffer_Cap* = 17 (Average) . . . 65
A.4 Results of energy consumption per zone of ZDG 66
A.5 Experimental results of ZDG for *Buffer_Cap* = 53 (Average) . . . 66
A.6 Variance calculation of EBDG 67
A.7 Variance calculation of Multihop-LEACH 67
A.8 Variance calculation of ZDG for *Buffer_Cap* = 17 68
A.9 Variance calculation of ZDG for *Buffer_Cap* = 53 68

List of Abbreviations

BS	Base Station
CDC-DSA	Cluster-based Data Collection scheme with Direct Sink Access
CH	Cluster Head
CSMA	Carrier Sense Multiple Access
DGR	Data Gathering Round
DGS	Dynamic Grid Switch
DN	Dissemination Node
DPs	Dissemination Points
EBDG	Energy Balanced Data Gathering
LEACH	Lower Energy Adaptive Clustered Hierarchy
MAC	Medium Access Control
MLDGT	Maximum Lifetime Data Gathering Tree
MST	Minimum Spanning Tree
SPs	Steiner Points
SPT	Shortest Path Tree
TBC	Tree Based Clustering
TDMA	Time-Division Multiple Access
TTTD	Two-Tier Data Dissemination
WSNs	Wireless Sensor Networks
ZDG	Zone Based Data Gathering
ZFDAT	Zone-based Fast Data Aggregation Tree
ZL	Zone Leader

List of Symbols

$E[\#CH]$	Expected number of cluster head per round
$p_i(t)$	Probability of a sensor to be a cluster head
$C_i(t)$	Cluster head indicator function
N	Total number of nodes in a wireless sensor network
N_z	Total number of nodes in a zone
r	Size of a zone
R	Communication range of a sensor node
(a, b)	Zone_ID
(a_{rz}, b_{rz})	Root zone_ID
M	Number of hops
$Buffer\_Cap$	Buffer capacity of a node
δ	Degree of a node
T	Shortest path tree
$E_{Tx}(k, d)$	Energy needed to spread k bits to an area of radius d
$E_{Rx}(k)$	Energy needed to demodulate k bits
d	Distance between transmitter and receiver
e_t	Baseline energy consumption level at the transmitter radio
e_r	Baseline energy consumption level at the receiver radio
k_0	Size of control packet
k	Size of data packet
$\epsilon_{fs}d^2$	Amplifier energy in case of free space model
$\epsilon_{mp}d^4$	Amplifier energy in case of multi-path model
E_{elec}	Electronics energy
d_0	Threshold distance
$d_{p,j}$	Distance between nodes
$d_{ZL,p}$	Distance between ZL and non-ZL node
$d_{z,z'}$	Distance between adjacent square zones
$E_{zf}(i)$	Energy consumption for zone formation in round i
$E_{sd}(i)$	Energy consumption for sending in round i
$E_{inter}(i)$	Energy consumption for inter-zone communication in round i
n_t	Total packets including incoming packets of a zone in round i
Q_t	Total number of cluster heads of a zone in round i
$E_{intra}(i)$	Energy consumption for intra-zone communication in round i
$E_{agg}(i)$	Energy consumption for data aggregation in round i
$E_{comm}(i)$	Energy consumption for data communications in round i
$E_{DGR}(i)$	Total energy consumption in a zone in data gathering round i

Acknowledgement

Life is a journey. I just finished an exciting ride. The happiness from this one will last for a long long time and I feel obligated to show my gratitude to the persons accompanying me during this wonderful trip.

Above all, I would like to express my heartiest gratefulness to the almighty Allah, without his divine blessing it would not be possible to complete the thesis successfully.

I would like to express my invaluable profound gratitude to my respective supervisor, Dr. Mahmuda Naznin, Associate Professor, Department of CSE, BUET for her constant and most sympathetic guidance to complete this thesis work. She encouraged and challenged me throughout the whole process. She always pointed me in the right direction when I was lost and supported me when I was on the right path. In addition to being my mentor, she has also been a good friend to me.

I am grateful to the head of the department of Computer Science and Engineering, BUET, for all time lab facilities which contributed greatly to my continuation of this thesis.

I wish to express appreciation to my colleague Mohammad Abdur Rouf, Assistant Professor, Department of CSE, DUET for providing many helpful comments.

I would like to thank my supervisor again for teaching me how to jump; how to fall; how to write; and how to fight.

Abstract

Wireless sensor network (WSN) technology is one of the popular applied technologies due to its various applications. Data collection or data gathering is a vital part of WSN. To make the network workable and to maintain the quality of services, energy management is also essential. To collect enough data, it is desirable to keep as many as sensors active which causes network energy failure. Trade-off between data collection and energy saving has made data collection scheme in WSN enough challenging. This research addresses the problem of energy limitations in data gathering. To handle the challenges of data gathering, we present an energy efficient data gathering approach ZDG which increases the network lifetime and packet delivery by partitioning the network into some zones. The tasks of data collection and aggregation are distributed among the nodes of the zones which resist the nodes from going to the bottleneck state by minimizing the data aggregation latency. Since the buffer capacity of a sensor node is limited, it is necessary to reduce data aggregation overhead also. To reduce the overload, we design a load balanced data collection scheme by dividing zone into clusters. We select some cluster heads in the zone to handle efficiently the buffer limitation. The clusters are on-demand clusters which helps to reduce energy consumption. In this thesis, we provide the theoretical analysis of energy consumption by our method. We simulate our technique with a popular sensor network simulator, OMNeT++. We compare our results with some popular zone-based data collection schemes for WSN. Simulation results show that our method improves the network lifetime and our method provides the higher throughput or packet delivery compared to those methods. We find that about thirty percent energy consumption is less in our method compared to the other popular methods. Our method has better packet delivery compared to those methods also.

Chapter 1

Introduction

1.1 Overview

Data gathering or data collection is an important part of any network research. Here two traditional solutions exist for Wireless Sensor Network (WSN). The first one is to store data locally in the network and to collect it manually. But, this is time consuming, labor-expensive, and impractical. The second solution is to wire up sensors, which makes their readings instantly available. Yet, this solution is also vulnerable, intricate and inflexible, costly, and infeasible. The advance in micro-computing has put forth WSN which consists of densely deployed sensors in a remote and harsh environment to detect natural phenomena from the environment and to send data to a remote sink [28, 30]. The wireless connectivity renders fast access to data and the deployment closed to phenomena make data collection better. The tiny size of the sensors allows of the deployment of a large number of nodes, so that a non-invasive monitoring solution with the high spatial resolution is achievable. This chapter discusses definition, applications and challenges of wireless sensor networks. The definition, categories and challenges of data gathering in wireless sensor networks are also described in this chapter.

1.2 Wireless Sensor Networks

Generally, the wireless sensor nodes are run with the battery. Sensor nodes are of low cost in general, very small in sizes and multi-functional [16]. They can sense physical or environmental conditions and can communicate in short transmission range. Fig. 1.1 [29] shows a typical sensor node structure, in which, MCU (or Micro Controller Unit) is the module that is responsible for controlling

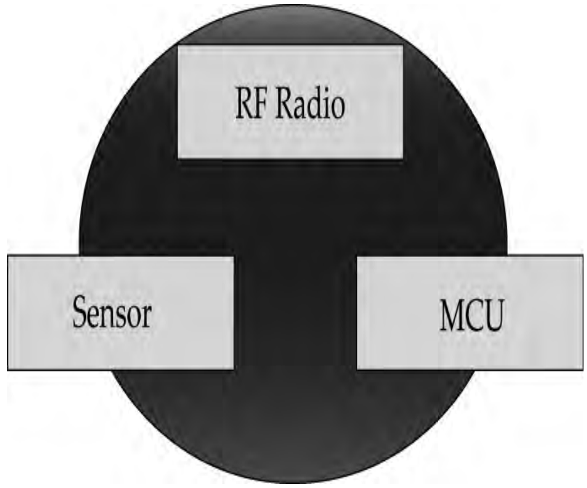

Fig. 1.1: A typical sensor node structure [29].

all activities of node and executing communication protocols. The sensor module includes sensors, the RF Radio module is responsible for wireless communication. Fig. 1.2 [11] shows the pictures of some of the latest commercial sensor nodes.

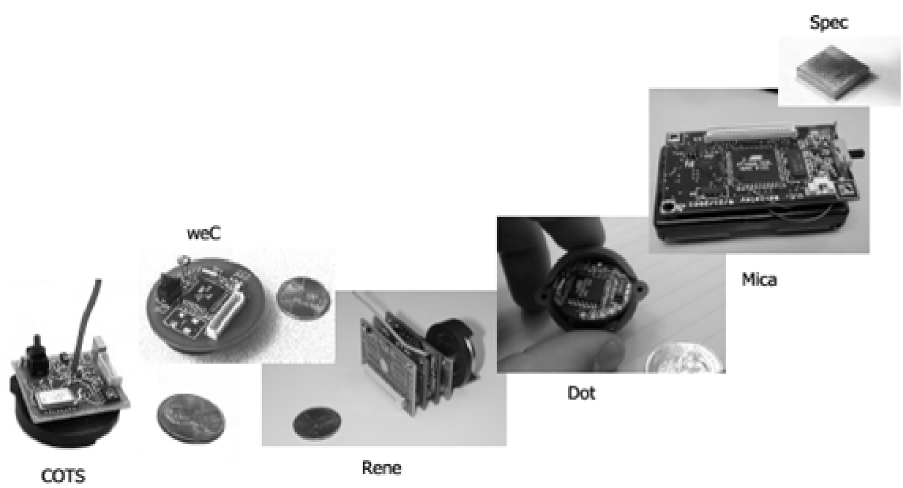

Fig. 1.2: Some current commercial sensor nodes [11].

1.3 Challenges in Applications of WSNs

Wireless Sensor Networks (WSNs) have many applications in real world. Because of WSN's salient features, it has some impacts on real applications.

1.3.1 Applications

Wireless sensor networks have significant impacts on some important military and civil applications, which are classified into three classes: data collection applications, surveillance applications, and object tracking applications [11]. For data collection from the environment we would want to collect data from different geographic locations. To monitor large systems such as assembly lines, shuttles, airplanes, the sensor network could help the engineers to monitor the system by periodically collecting data about the systems' modules.

In general, monitoring applications can be subdivided as one-to-one and many-to-one communication paradigms [22]. In the one-to-one communication paradigm, sensor nodes generate events and report to a destination on demand basis. In the many-to-one paradigm, each sensor node periodically performs measurements, possibly pre-processes, and finally reports to a central node. The many-to-one paradigm is frequently adopted in data gathering scenarios, such as environment or agricultural monitoring. Wireless sensor networks have been deployed in a variety of settings to assist researchers with detailed information. In the domain

Fig. 1.3: A wireless sensor network for data collection in a crop field [11].

of environmental monitoring, sensor networks are currently employed to record the ecophysiology of forests, gain understanding of the turbulent sub-grid scale physics near the snow-atmosphere interface, measure soil moisture tension for irrigation management, or to survey glacier displacement [14]. The agricultural monitoring has recently been employed to observe climate change, soil test and pest controlling. Fig. 1.3 [11] shows a model of using a sensor network to collect information about the soil from a crop field. A particularly challenging example is that of tideland monitoring, where sensor nodes are placed on off-shore in order to collect sampling data, and to report their readings during ebb tide [3]. Natural habitat studies use sensor networks to observe seabird colonies or to track long term animal migrations.

Surveillance applications deploy sensor nodes by placing them at predefined locations. All nodes continuously monitor the environment, and when there is an anomaly, they report to the base station. Indoor navigation [5] deploys many sensor nodes at some fixed locations inside a building to monitor the movement of objects in the building.

Fig. 1.4: A scenario of tracking the enemy using sensor networks [25].

Object tracking is one of the most important applications of wireless sensor networks. For example, WSN is being used to track packages during the shipping. In a tactical system, the network could detect and track enemy movement. Also, the wireless sensor network may replace humans in patrol missions within a dangerous environment or over challenging terrains. Fig. 1.4 [25] is a scenario of applying sensor networks in tactical system.

In the industrial sector, sensor networks have the potential to play a significant role in facility management, process monitoring, and security applications. Usually, the large chemical and refining plants are assessed by sensor systems. Furthermore, wireless sensor networks are also being used into medical and disaster reliefs. Some other applications are green house monitoring, air/water pollution detection, civil engineering, nuclear, biological and chemical attack detection etc.

1.3.2 Challenges

The applications mentioned earlier generate continuous and sometimes periodic data at low rate. Sensor sampling intervals of several minutes are sufficient to provide adequate data resolution. Furthermore, incurred network delay is of minor concern as the collected information is often analyzed offline [28]. On the other end,some applications such as structural monitoring or the observation of volcanic activity generate data at a very high rate. In these systems, it is desirable for a sensor network to sleep as much as possible and wake up very quickly upon any event triggering.

Due to the small sizes and relatively low cost, wireless sensor nodes are available in the market with low computational power. Moreover, they suffer from the limitations of small memory, buffer capacity, wireless transmission range, and bandwidth. Furthermore nodes are battery-powered. Since battery replacement is usually not feasible in a large, remote network or even impossible in harsh environments, conservation of energy is a crucial point in order to maximize the network lifetime.

As a result of the restrictions concerning small memory and low computational power of sensor nodes, applications and protocols' layout must be done carefully. Much data cannot be buffered due to the memory constraints and again continuous data processing and computing are also energy consuming. Therefore, the tradeoff between conservation of memory and energy consumption is required.

As the radio transmission is the major energy consuming process, sensor nodes go to the *sleep mode* when they are not working, in which they switch off the radio and sensing devices. The fraction of time a node actually spends outside a sleep mode is called *duty cycle*. Therefore, it is desirable to have a selection policy for the sensors when to go to in sleep mode.

The most obvious source of energy dissipation is that of *signal interference* [22]. Here, a signal results from the packet transmission. If a node receives signals from two or more different senders at the same time *signal interference* occurs. If at least two of the packets are destined to that receiver, this is called a *collision*. An attempt to avoid interference is the application of contentionbased, also called *Carrier-Sense Multiple Access* (CSMA), protocols that do a carrier sense before starting a transmission [2]. If the sensed signal is below the carrier-sense threshold, the sender assumes a clear channel and commences transmitting. If not, a new carrier sensing starts after a backoff. Two nodes which are outside of each others' communication ranges can start transmissions at the same time. If at least one of the intended receivers is inside the communication range of both of the senders, interference occurs.

Idle listening is another mode of a sensor node when the radio channel consumes more energy than the sensor is in the sleep mode [21]. Additionally, listening on the channel leads to *overhearing*. Here, nodes receive (overhear) packets not destined to them, which causes the wastage of energy due to useless reception and packet processing. Reception and transmission draw approximately the same power. Both problems are attenuated by scheduled *Medium Access Control* (MAC) protocols, which is responsible for controlling the radio; *Time-Division Multiple Access* (TDMA) [3, 21]. In TDMA, time is divided into some transmission slots, and slots are assigned to nodes. If wireless communication is required, the efficient usage of the radio is mandatory.

1.4 Challenges in Data Gathering

1.4.1 Data Gathering

Data gathering is the process of collecting sensed data from the sensor nodes, and transmitting to the base station/data sink [14]. Data gathering algorithms are categorized according to the network architecture. Data gathering strategy can be classified mainly into three categories: *Grid Based Approaches*, *Tree Based Approaches*, and *Cluster Based Approaches* [14].

In a *grid based approach*, the network is mapped to a virtual grid where sensors closed to the grid points are elected as the dissemination or forwarding nodes. A sink connects to the nearest grid point in the local grid, and uses the overlay network composed of dissemination nodes to obtain data from the source nodes. In case of mobile sinks, when any sink moves, it connects to a new local grid for collecting data.

Tree based approach uses mainly a tree like structure for data gathering. In tree based approach, each sink has a data gathering tree rooted at itself. This tree can be a locally built the minimum spanning tree [4] including all the sensor nodes, or a simple reverse multicast tree with data sources as leaves.

Cluster is the most favored structure for date gathering. In cluster based approach, sensors are grouped into clusters, and each cluster has a node elected as cluster head. In a cluster, cluster members send their readings to the cluster head, which processes its received data, and report the aggregated data to the sinks.

1.4.2 Challenges

There are many challenges involved in data collection using WSNs. First of all, the deployed sensors may need to cover the full area. And to acquire data accurately, sensors may be required to be placed at some specific locations. In addition, different types of data (temperature, light, vibration) may be obtained by different sensors with different sampling rates [28, 31]. These issues may cause unbalanced energy consumptions over a WSN and significantly may shorten the network lifetime. In addition, since it is required to deliver data to the base station without any information loss, the data aggregation/fusion is also very hard.

More specifically, in many-to-one traffic pattern, if not carefully handled, high unbalanced and inefficient energy consumption occur in the network. For example, the *energy hole problem*, where sensor nodes closed to the base station are depleted quickly due to the heavy traffic, and the network is disconnected from the base station [17]. One possible solution to alleviate this type of issues and to pursue the maximum energy saving is to use a mobile collector that proactively moves around and collects data in the sensing field [26, 28]. But, the low velocity of the mobile collector incurs a long latency in data gathering, which does not meet the delay requirement of different time-sensitive applications.

In addition, sensors used in data collection collect a huge volume of data and the data may be of different types. These sensors work at their own sample rates specified by the applications, and these rates may be different from each other. Such differences lead to the different transmission rates to relay data from the different types of sensors, which may further cause the unbalanced traffic load and eventually the energy consumption. Unbalanced energy consumption causes performance degradation.

Moreover, network operation may be disturbed due to the frequent network connectivity failure. Data collection operation must be continuous by handling the network connectivity problem. Furthermore, *throughput* i.e, the number of packets received by data sink/base station is a vital challenge in data gathering within the limit of energy consumption and connectivity failure. The data must be available to the sink with the minimum data gathering latency. To collect enough data, it is desirable to keep as many as sensors active which causes network energy failure. On the contrary, to keep most of the sensors inactive reduces data collection. The problems arising in the field of data gathering in WSN need to study thereby always aiming at energy efficient solutions. The physical and medium access layers design, topology control, and routing techniques all are to be optimized to comply with the stringent hardware limitations of the sensors.

In this thesis, we address the issues of data gathering technique with a tradeoff between the energy saving and the throughput. In our scheme, we fix a goal to develop a strategy for data transmission with the energy saving. We provide a better packet delivery scheme with a good network life time.

1.5 Thesis Outlines

The remaining chapters of this thesis are organized as follows. Chapter 2 reviews related research work. Chapter 3 describes the problem domain and the details of our methodology. The performance metrics and related mathematical analysis are provided in Chapter 4. Chapter 5 provides the experimental results and performance analysis. Finally, in Chapter 6, conclusions are drawn with making a reference to some possible future directions of this research.

Chapter 2

Related Work

2.1 Overview

"When we mean to build, we first survey the plot, then draw the model."

- William Shakespeare, *Henry IV Part II*, Act *I*, Scene iii.

Many approaches have been proposed to tackle the challenges of data gathering. The strategies provide improved facilities in data collection schemes. But still they suffer from some major limitations. In this chapter, a brief analysis of recent related work is carried out to find their strengths and weaknesses.

2.2 Grid Based Approaches

This section describes the related grid based approaches of data gathering.

2.2.1 Static Grid and Dynamic Grid Approaches

In [12], Khan *et al.* proposed an analytical model for data gathering in grid based WSNs. Two algorithms, static grid and dynamic grid algorithms are discussed in [12]. In static grid algorithm, source forms a uniform grid, and it is one of the nodes of the grid. Whenever the sink is in need of data, it sends the query towards the nearest cell head. Once the query reached the cell head, it transmits the query along the grid towards the source. On reception of the query by the source, it sends back the data to the corresponding sink. Again, it uses the grid nodes for data forwarding. In order to forward the traffic using grid nodes, it uses 'X-Y routing' technique. In 'X-Y routing' mechanism, the node which has to forward the packet based on geographical coordinates. First, the packet is forwarded in x-direction until x-coordinate of the current node equals to the x-coordinate of the destination. Then, it is forwarded in y-direction, until the y-coordinate of the current node equals to the y-coordinate of the destination. Thus, packet reaches the destination.

In dynamic grid algorithm, backbone grid is changed after a specific interval of time. In this algorithm, first of all the network nodes are divided into more than one grids. Each grid is located at a uniform distance from the neighboring grid. Thus, the role of backbone communication is uniformly rotated towards all the grids. The nodes nearest to the crossing points of the grid lines are chosen as the head for the back-bone communication. In order to send the queries by a sink, it uses the grids formed by the different nodes for different queries. Thus, this process rotates the query generation on each of the virtual grid, and shares the load of packet forwarding.

In order to send the query from sink to the source, the query is first sent towards the local cell head which is active at the present moment. Once it is reached at the cell head, the backbone grid forwards the query towards the destination cell where the source is located. The cell head of source cell forwards the query towards it. On receiving the query, the source analyzes it and send the data towards the sink. The data forwarding follows same mechanism as that of query forwarding. First of all, it is sent towards the sources cell head. The sources cell head forwards the data towards the sinks cell head. Once the data is reached towards the sinks cell head, it is forwarded towards the sink. In order to forward the query/data on the grid, 'X-Y routing' mechanism is used.

Although both the structures are easy and simple but due to excessive data gathering points, many nodes need to perform aggregation. Moreover, the schemes [12, 36] use 'X-Y routing' mechanism which causes more energy consumption due to participation of many nodes in the horizontal and the vertical data forwarding. The cell nodes closer to the sink consume more energy than others. It may arise data transmission/reception problem due to the creation of network cuts between the source and the sink.

2.2.2 Dynamic Grid Switch Approach

In order to solve the load imbalancing problem in [12], Zu-jue *et al.* introduced DGS [36], a Dynamic Grid Switch multi-sink data gathering mechanism which constructs grid based on the Two-tier Data Dissemination (TTDD) [34]. The construction of the grid includes three steps:

• Select position of intersection point. First, initialize the nodes on the lattice according to their own values and determine the coordinates in the grid from their recent location.

• Select the intersection node nearby the intersection point. Within a radius of the size of the region, the nodes compare the energy between themselves and elected the next intersection node with the biggest ratio of energy of the node to the distance between the node and intersection point. The ratio is expressed as: $Max(node\_energy/Distnode, intersect)$.

• Contact between intersection node. The selected intersection node makes contact with the surrounding neighboring nodes to attain mutual recognition of each other's position, thus the grid is built completely.

After the establishment of the grid, when event occurs, the node that captures the event is called as a *source node*. The source node looks up the nearest intersection and send the information. The intersection node which follows in adjacent geocasts this information until the event message which includes event types and occurrence locations reaches the entire sensor network. While a sink node sending a query, it does the local flooding to determine whether there is information needed to retrieve. If so, it sends back to the sink node according to the information stored, if not it can immediately notify the sink. In this method, the grid switch dynamically updates the transmission path in order to achieve load balancing.

DGS adds even driven mechanism to dynamically adapt the whole network energy status. DGS also supports mobile sinks in such environment. All intersection node switches seamlessly in accordance with the rules, thus effectively avoids excessive depletion of intersection node in a single regional area where sudden or persistent data queries may happen.

DGS suffers from the limitations of energy consumption due to the horizontal and vertical forwarding of the data which causes participation of many nodes in data forwarding.

2.2.3 Steiner Points Protocol

Steiner points protocol [15] improves the performance of grid routing protocols discussed in [12, 36]. It delivers the low energy consumption and the high transmission efficiency features. In steiner points protocol, once the sensor nodes are deployed in the sensor field, the sink node starts to construct the grid structure. The sink divides the plane into a grid of cells. Cross-points of the grid are the *Dissemination Points* (DPs). The size of the cells, denoted as α, is determined by the sink such that DPs are not within direct transmission range. The sink is the first DP. Knowing its own position and the size of each cell, the sink is able to send a data request (in the form of a data announcement message) to each adjacent DP in the grid. For a sink at location $L_s = (x, y)$, dissemination points are located at $L_p = (x_i, y_i)$, such that:

$$\{x_i = x + i.\alpha, y_j = y + j.\alpha; i, j = \pm 0, \pm 1, \pm 2, ...\} \tag{2.1}$$

The positions of X and Y depend on the length of square, so the *Steiner Points* (SPs) are located at $L_{sp} = (x_i, y_j)$ such that:

$$\{x_i = x + i.\frac{\sqrt{3}}{6}\alpha, y_j = y + j.\frac{\sqrt{3}}{6}\alpha; i, j = \pm 0, \pm 1, \pm 2, ...\} \tag{2.2}$$

Simple geographic forwarding is used to reach these locations. Upon receiving the data-announcement message, the closest known sensor node to each DP and SP promotes itself to become a *Dissemination Node* (DN), and DN records which DP or SP it belongs to. Then, the DN forwards the data announcement message to each of its adjacent DPs and SPs, except the point from which the data announcement message was received. These actions are repeated as the data announcement message propagates and DNs are chosen throughout the sensor field.

Any node that is within the target region of a received $QUERY$ message stores the appropriate routing information, and starts to send the sensed data (in the form of $DATA$ message) to the sink. The routing information contains the appropriate upstream DNs through which $DATA$ messages are forwarded. DN finds the appropriate path to transmit $DATA$ message depending on which DP or

SP it belongs to. If DN belongs to SP, then it chooses the upstream path along the hexagonal structure. However, there are two different paths those can be chosen if DN belongs to DP. If the location of DP, denoted by $L_p = (x_i, y_j)$, is parallel or vertical to the location of sink node, denoted by $L_s = (x, y)$, then DN forwards the data along the upstream path on the grid cell. Otherwise, DN forwards data along the upstream path on the hexagonal structure. Having established paths to the sinks through DNs, data forwarding can occur. The forwarding procedure at a DN continues till the $DATA$ message stops at the sink node.

Creating virtual grid structure based on square steiner trees can handle black hole problem [17], but the addition of new data gathering points forces extra aggregation latency. Besides the cell nodes closed to the sink consume more energy than the others.

2.3 Tree Based Approaches

This section describes the related tree based approaches of data gathering.

2.3.1 2-Approximation Double Tree Algorithm

In [30], Weng *et al.* presented double tree algorithm, for raw and encoded data gathering. The algorithm is named $MST + SPT$ (Minimum Spanning Tree + Shortest Path Tree). It uses the minimum spanning tree to be the tree for transmitting raw data, and uses shortest path tree to transmit the encoded data. Detail of $MST + SPT$ is discussed as follows:

• At first, MST and SPT are computed, keeping the sink as the root of MST.

• Then the sink broadcasts its data rate value R_{sink} to its all children nodes in MST. When node v is receiving a data rate value R_u from its parent node u in MST, node v encodes its own data R_v using R_u, transmits its encoded data r_v to the sink along the shortest path and,

• Finally, node v broadcasts its data rate value R_v to its all children in MST. The sink reconstructs all original sensing data of all nodes by performing recursive decoding, based on the encoded data gathering from all of the nodes.

Double tree algorithm suffers additional overhead for building the two distinct trees.

2.3.2 Maximum Lifetime Data Gathering Tree (MLDGT)

Maximum Lifetime Data Gathering Tree (MLDGT) [31] avoids building the trees [30]. MLDGT considers a connected sensor network G of N nodes. Each node monitors the environment and periodically generates a small amount of data. To gather the data from the sensor nodes, it needs to construct a tree-based topology after node (re)deployment. Essentially, the solution starts from an arbitrary tree, and iteratively makes "improvements" by reducing the degree of the bottleneck nodes, i.e., nodes with a short lifetime, at each step.

Consider a tree T and an edge (u, v) that is not in T. Let V_1, V_2, and V_3 be the sets consisting of bottleneck nodes, nodes closed to become bottleneck node and "safe" nodes respectively, and a unique cycle C be generated when (u, v) is add to T. If there is a bottleneck node $w \in V_1$ in C, while both u and v are in V_3 (safe nodes), then (u, v) is add to T and delete one of the edges in C incident on w. This reduces the degree of the bottleneck node w by one. This modification is called an improvement, and say that w benefits from (u, v). MLDGT uses this method as a building block to increase the network lifetime.

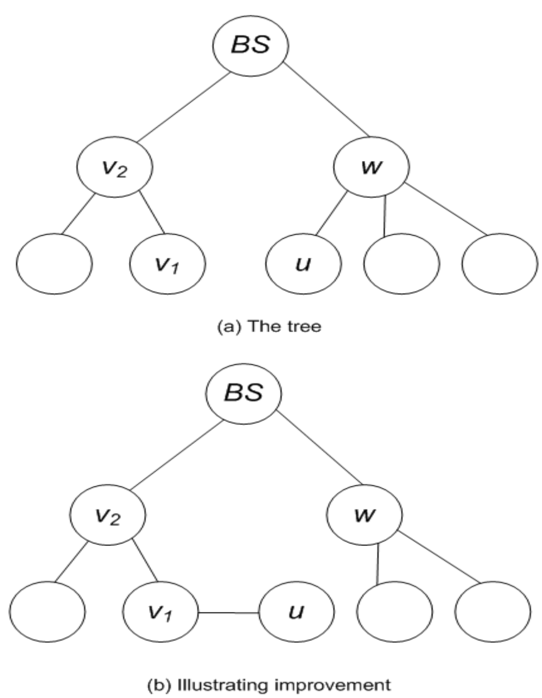

Fig. 2.1: The notion of improvement [31].

For the above example, if either u or v or both are in V_2, the above modification turns u or v or both into bottleneck node(s). Thus while reducing the degree for one bottleneck node, it produces additional bottleneck node(s). This is undesirable and we say that w is blocked from (u, v) by u (or v or both). A node is blocking if it is in V_2.

Fig. 2.1 [31] depicts the process of MLDGT. According to the above definition, if w is a bottleneck node, v_2 is a blocking node and all other nodes are safe. It is possible to add (u, v_1) and delete (w, u). This is an improvement as it reduces the degree of the bottleneck node w. In contrast, adding (u, v_2) and deleting (w, u) do not prolong the network lifetime, because doing so produces another bottleneck node v_2 while reducing the degree of w.

In MLDGT, reducing the degree of nodes may cause other nodes to be bottleneck. Meanwhile after finding a bottleneck node, it is difficult to reduce the degree of nodes when composite components exist. Definitely, this update process is time consuming.

2.4 Cluster Based Approaches

This section describes the related cluster based approaches of data gathering.

2.4.1 LEACH and Its Variants

One of typical cluster based approaches is Lower Energy Adaptive Clustered Hierarchy (LEACH) [8–10], where nodes are selected randomly as cluster heads. A scenario of cluster formation is shown in Fig. 2.2 [8–10]. In LEACH, the nodes

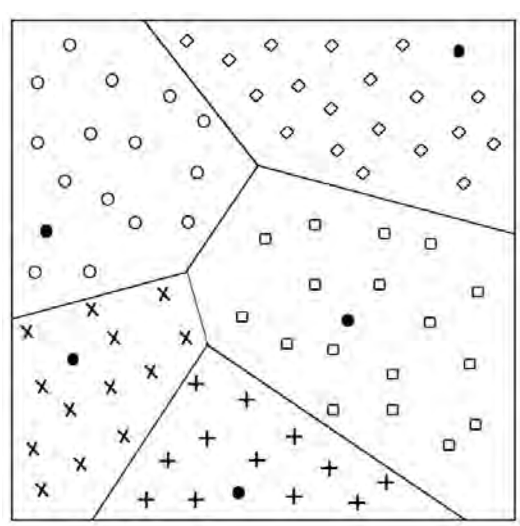

Fig. 2.2: An example of cluster formation [8–10].

organize themselves into local clusters, with one node acting as the cluster head. All non-cluster head nodes transmit their data to the cluster head, while the cluster head node receives data from all the cluster members, performs signal processing functions on the data (e.g., data aggregation), and transmits data to the remote base station (BS). Therefore, being a cluster head node is much more energy intensive than being a non-cluster head node. If the cluster heads were chosen a priori and fixed throughout the system lifetime, these nodes would quickly use up their limited energy. Once the cluster head runs out of energy, it is no longer operational, and all the nodes that belong to the cluster, lose communication ability. Thus, LEACH incorporates randomized rotation of the high energy cluster head position among the sensors to avoid draining the battery of any one sensor in the network. In this way, the load of being a cluster head is evenly distributed among the nodes.

The operation of LEACH is divided into rounds. Each round begins with a set-up phase when the clusters are organized, followed by a steady-state phase when data are transferred from the nodes to the cluster head and on to the BS, as shown in Fig. 2.3 [8–10]. Each sensor i elects itself to be a cluster head at the

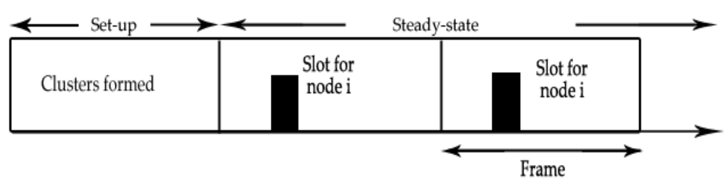

Fig. 2.3: Time line showing LEACH operations [8–10].

beginning of round $r+1$ (which starts at time t) with probability $p_i(t)$. $p_i(t)$ is chosen such that the expected number of cluster head nodes for this round is k. Thus, if there are N nodes in the network, the expected number of cluster head per round-

$$E[\#CH] = \sum_{i=1}^{N} p_i(t) * 1 = k \qquad (2.3)$$

Ensuring that all nodes are cluster heads the same number of times requires each node to be a cluster head once in N/k rounds on average. If $C_i(t)$ is the indicator function determining whether or not node has been a cluster head in the most recent $(r \ mod \ (N/k))$ rounds (i.e., $C_i(t) = 0$ if node i has been a cluster head and one otherwise), then each node should choose to become a cluster head at round with probability

$$p_i(t) = \begin{cases} \frac{k}{N-k*(r \bmod \frac{N}{k})}, & \text{if } C_i(t) = 1 \\ 0 & \text{, if } C_i(t) = 0 \end{cases} \tag{2.4}$$

Therefore, only nodes that have not already been cluster heads recently, and which presumably have more energy available than nodes that have recently performed this energy-intensive function, may become cluster heads at round $r + 1$.

But, the major limitation of LEACH is that all cluster heads need to communicate directly with the sink which causes channel overhead at the sink. Moreover, in the random selection of the cluster head, there may not exist any cluster head in some region of the network which causes failure of LEACH as LEACH takes only 5% of the total nodes as the optimal number of cluster heads. Each cluster head directly communicates with the sink no matter the distance between the cluster head and the sink is far or near. It consumes lot of energy if the distance is far.

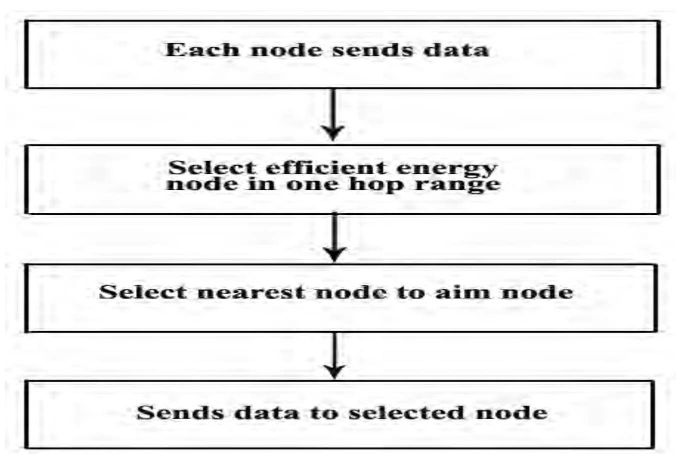

Fig. 2.4: Routing of Multihop-LEACH protocol [32].

A further modified LEACH protocol (denoted as Multihop-LEACH protocol) [32] which selects optimal path and adopts multi-hop between cluster head and sink is presented. Its multi-hop routing algorithm within one round is shown in Fig. 2.4.

The scheme works as follows: First, multi-hop communication is adopted among cluster heads. Then, according to the selected optimal path, these cluster heads transmit data to the corresponding cluster head which is nearest to sink. Finally, this cluster head sends data to sink. Multihop-LEACH protocol is almost the same as LEACH protocol, only makes communication mode from single hop to multi-hop between cluster heads and sink.

Other variants of LEACH [1, 8–10, 18, 24, 33] including Multihop-LEACH [32] describe improved approach for cluster formation and cluster head selection. But, all of them like LEACH assume that all nodes have enough power to communicate directly with the sink which is not practical as sensors do not have long communication range usually [17].

2.4.2 Cluster-based Data Collection scheme with Direct Sink Access (CDC-DSA)

A Cluster-based Data Collection scheme with Direct Sink Access (CDC-DSA) is discussed in [16]. In CDC-DSA, stationary sensors directly communicate with cluster heads (CHs) and therefore form one-hop clusters. CDC-DSA generates clusters periodically in a random manner using any general clustering algorithm. In each sink-initiated round of data collection, CHs based on TDMA receive data samples from their cluster members, and after data aggregation, they contend to access the reach-back channel to transmit the aggregated data to the data sink.

Introducing direct-sensor sink communication shifts the expensive task of routing, maintaining routing tables, and network control toward data sinks. The scheme maintains same TDMA schedule for several round of data collections, thus the cost and delay for cluster formation and TDMA schedule setup is negligible.

The major drawback of CDC-DSA is that it suffers from the direct communication problem. Moreover, it provides low data gathering rate because only one CH can transmit data at a time to the data sink. Rest of them remains in sleep mode. The CDC-DSA protocol works, if only a fixed rough estimation value or the total number of CHs is known a priori.

2.4.3 Novel Energy Efficient Cluster-based Algorithm

Another cluster-based data gathering method for WSNs is discussed in [17] which improves the performance of LEACH [8–10]. In the proposed method, cluster heads send data to the base station by multi-hop communication. The algorithm is divided into five steps:

a) **Counting hop number:** In this step, broadcasting from the base station (BS) is used to count the hop number for each sensor node to the BS. Broadcasting is used, because the same broadcasting message from the BS could reach a sensor node through many routes. When the broadcasting messages are received by a sensor node, the sensor node uses a routing table in its memory to cache all the routes from the BS to itself.

b) **Hierarchicalizing sensor nodes:** To hierarchicalize sensor nodes into different layers to set the level of sensor nodes, use the hop number from the BS to each sensor node. Set sensor nodes which communicate with the BS via the same hop number into one layer.

c) **Clustering in the system:** In this step, the sensor nodes are clustered into clusters, and here are the points as follow.

• Use the concept of LEACH Algorithm [8–10] to decide cluster head (CH) nodes in the system.

• For recognize each CH node in their layers, mark them as:

$Level_1 CH_1, Level_1 CH_2, ...$

$Level_2 CH_1, Level_2 CH_2, ...$

• The other non-CH nodes join a cluster using 1-hop communication, depending on the strongest transmission signal.

• Because there could be sensor nodes that could not find a CH node in their transmission range, so they elect themselves as CHs.

d) **Transmission and scheduling in clusters:** A CH node schedules all sensor nodes in its cluster with TDMA (Time-Division Multiple Access) medium access control to avoid collision.

e) **Selecting transmission routes:** Establish routes with routing algorithms for CH sensor nodes data transmission, which concerns:

• First, minimize the hop number.

• Second, maximize the energy in selected routes.

• Third, establish routes for transmission with the method addressed above, for all CH nodes and non-clustered sensor nodes.

• Last, discover the alternative sensor node (ASN) while transmitting data.

The approach tries to balance the energy usage in the network by employing alternative sensor nodes but it suffers from the self-induced black hole problem where the nodes near the sink die earlier.

2.4.4 Tree Based Clustering (TBC)

In [13], Kim *et al.* proposed a tree-based clustering (TBC) with a better performance than that in [17]. In TBC, the nodes form a tree with the root as cluster head, while the height of the tree is decided based on the distance of the member nodes to the cluster head. The cluster head is decided by the same way as LEACH [8–10]. After the clusters are formed, each cluster constructs a tree with the member nodes where the cluster head becomes the root. The tree construction consists of two steps. The first step is the determination of the tree level of each member node in the cluster, and the second one is the formation of the tree based on that. The distance of a node to the root serves as the basis for determining the level in the cluster. Let d_{max} denotes the distance of the node farthest from the cluster head and α, the number of levels decided according to the size of the network. Then the average data transmission distance between two adjacent levels of the tree, d_a can be calculated as follows.

$$d_a = \frac{d_{max}}{\alpha} \tag{2.5}$$

Let $L(i)$ denotes the level of *node* i, N_i. In order to construct a tree in the cluster, the cluster head decides the parent node of each member node to which it sends the data. To make the decision, the set of candidate parent nodes of N_i, PS_i is defined as $PS_i = \{N_j \text{ for which } L(j) = L(i) - 1\}$. Thus, PS_i consists of the neighboring nodes of N_i locating at $L(i-1)$. After constructing the tree, the cluster head creates a TDMA schedule indicating when each node in the cluster can transmit the data. This schedule is broadcasted to the nodes in the cluster. The member nodes receiving the schedule message send the data to the parent node in the allocated transmission time during the data transmission phase. In TBC scheme, the ($dest\_id$) field is included in the schedule sent to the member nodes, which contains the ID of the node to which the data is transmitted. Each node refers to this field to send data during the allocated time.

Though TBC balances energy among the nodes, it suffers from the similar limitation like LEACH. Moreover in TBC, when the number of levels is small, the average transmission distance becomes too long and data transmission consumes more energy, and when the number of the levels is high, excessive packet forwarding is required and energy is wasted.

2.4.5 Zone-based Fast Data Aggregation Tree (ZFDAT)

In [20], Quan *et al.* proposed an optimal link scheduling zone based approach to minimize the aggregation time by given variable length of time slots for all links in the zone-based fast data aggregation tree (ZFDAT). In ZFDAT, clusters are formed as like in LEACH [8–10] and a number of circular zones are created inside a cluster where the cluster head (CH) is the local control center to coordinate the data transmission in the cluster. The ZFDAT strategy works as follows:

- Initially, each CH announces its own location information to reachable all sensor nodes using non-persistent carrier sense multiple access (CSMA) MAC protocol with maximum transmit power(P_{max}) of a sensor node as like in LEACH [8–10].
- After receiving the announcement, each non-CH node decides to which cluster it belongs by comparing the received signal strength from all neighboring CHs. Then each non-CH node estimates minimum hop distance between itself and its CH by using the received power (P_{rx}) from the CH and a sensor nodes transmission range(T_r).
- Each non-CH node has to inform the CH to claim itself to be member of the CH. Each non-CH node broadcasts its minimum hop count toward CH to its neighboring nodes. Only the neighboring nodes which have smaller hop count than receiving hop count sends feedback message.
- Upon receiving the Joint-REQ messages, each intermediate node can deduce how many children it has, while a node that does not receive any Join-REQ messages thinks itself as a leaf-node.

The drawbacks of ZFDAT are - i) clock synchronization is required which is very difficult in WSNs and ii) all nodes in intra-zone must have to participate in aggregation results excessive energy consumption.

2.4.6 Cocentric Data Aggregation Model

A further improved zone based approach is discussed in [27] which considers the location of the base station, and divides the whole sensor network into several concentric and hierarchical zones. Refer to the location of the base station, each zone is also divided into some areas. Then, each node in the sensor network can be assigned its own zone from base station by broadcasting a specific packet from low power level to high. The distribution of the zone is formed like a concentric circle. The interval of zone may be various according to the setting values of the base station. The values are depended on various parameters such as the density of the sensor networks, the number of nodes, or the location of the base station. After the process of zone assignment, each node has a serial number of its own zone. Nodes send their data to head by constructing chain. The chain construction in each zone is started at the farthest node of the zone from the base station by using the greedy algorithm.

The drawback in [27] is that all nodes in a chain constructed by using a greedy algorithm, have to participate in forwarding data which ultimately increases the overall network energy expenditure.

2.4.7 Energy Balanced Data Gathering (EBDG)

In order to balance energy consumption between nodes, maximizing network lifetime and eliminating black hole problems of the existing protocols, Energy Balanced Data Gathering (EBDG) [35], a zone based protocol, divides the circular monitoring area A into n concentric coronas $C_1, C_2, ...C_n$ centered at the sink with the same width r ($r = \frac{R}{n}$), as shown in Fig. 2.5 [35]. For any node $u \in C_i$, it

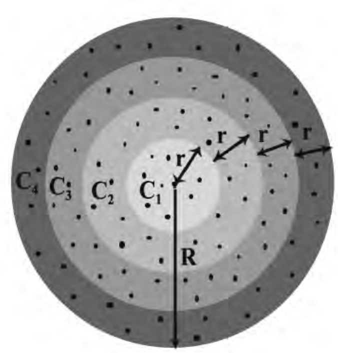

Fig. 2.5: Illustration of network division in EBDG [35].

forwards its data to a neighbor in C_{i-1} when hop-by-hop transmission mode is used, and transmits its packets directly to the sink when direct transmission mode

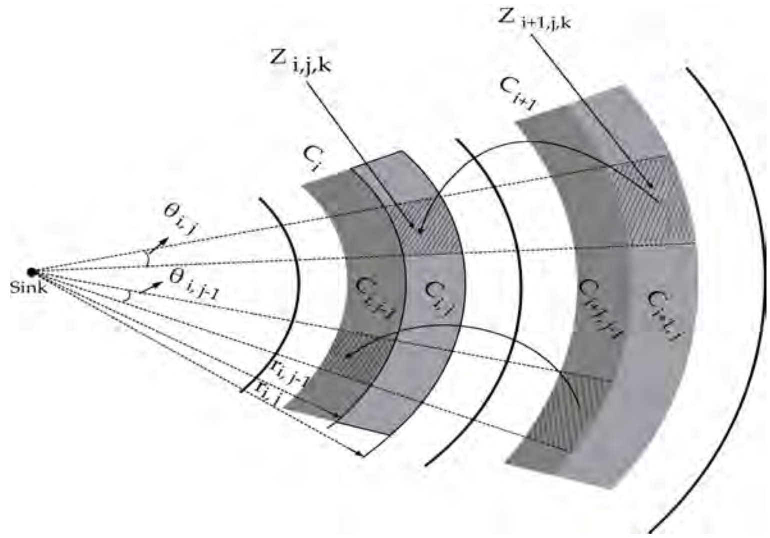

Fig. 2.6: The partition of coronas C_i and C_{i+1} in EBDG [35].

is used. Balanced energy consumption is achieved by optimally distributing the amount of data for hop-by-hop and direct transmissions at each node.

The basic idea of the zone-based routing scheme is explained as follows: each corona is divided into subcoronas and each subcorona is further divided into zones, as shown in Fig. 2.6 [35].

There is a one-to-one mapping between the zones in two adjacent coronas. Data communication is performed between nodes in two corresponding zones. The objective of EBDG is to design an optimal zone division scheme so that the amount of data received by nodes in each corona can be balanced. The detailed zone division approach is described as follows.

Consider any two adjacent coronas C_i and C_{i+1} where nodes in C_{i+1} forward their data to nodes in C_i when hop-by-hop transmission mode is used. C_{i+1} is termed as the source corona and C_i is termed as the destination corona. As shown in Fig. 2.6 [35], the source corona C_{i+1} is divided into w subcoronas $C_{i+1,1}, C_{i+1,2}, ...C_{i+1,w}$ with equal width $\frac{r}{w}$ and the destination corona C_i is divided into w subcoronas $C_{i,1}, C_{i,2}, ...C_{i,w}$ with width $\Delta_{i,1}, \Delta_{i,2}, ...\Delta_{i,w}$ respectively. Each subcorona $C_{i,j}$ is further divided into h_j equal-size zones $Z_{i,j,1}, Z_{i,j,2}, ...Z_{i,j,h_j}$ in such a way that $Z_{i,j,k}$ and $Z_{i+1,j,k}$ are the regions that $C_{i,j}$ and $C_{i+1,j}$ overlap with a circular sector centered at the sink with radius R and central angle $\theta_{i,j} = (\theta_2 - \theta_1)$, where $\theta_1 = \frac{2\pi(k-1)}{h_j}$ and $\theta_2 = \frac{2\pi k}{h_j}$ (θ_1 and θ_2 are the angles of the two radii of the sector in polar coordinates). Each destination zone covers λ nodes where λ is a user-defined parameter. Obviously, all destination zones have the same area size, and the same for the source zones.

In the zone-based routing scheme, $Z_{i+1,j,k}$ is referred to as a source zone, and $Z_{i,j,k}$ is referred to as the corresponding destination zone for $Z_{i+1,j,k}$. Each node u in $Z_{i+1,j,k}$ forwards its data to all nodes in $Z_{i,j,k}$ with equal probability. To achieve this, each node maintains the location information of the nodes in its destination zone. Since the number of nodes in each destination zone is small, this scheme does not add too much complexity to the zone-based routing scheme.

In EBDG, load balancing is achieved by sending data to the adjacent circles. Besides in this method, all of the nodes must have to participate equally to transmit data which causes unnecessary energy consumption. Furthermore, problem of empty zone arises.

Therefore, it is very challenging to find a solution for maximizing data delivery with the minimization of the energy consumption.

Chapter 3

Problem Domain

3.1 Overview

A well designed energy efficient data gathering approach is the crucial need now-a-days to achieve an improved network lifetime of WSNs. This chapter describes a zone based data gathering (ZDG) approach which provides energy efficiency with a better packet delivery.

3.2 System Model

3.2.1 Preliminaries

We consider a WSN with N randomly deployed nodes that are divided into a set of square areas called *zones* as shown in Fig. 3.1. The length of a square zone is set to $r = \frac{R}{\sqrt{2}}$, where R is the transmission range of a sensor node. Each node has an ID called *node_id*. The ID of a zone called *zone_ID* is denoted by (a, b). *zone_ID* (a, b) of a node can be estimated from its position (x, y) as: $a = \left\lfloor \frac{x-x'}{r} \right\rfloor$, $b = \left\lfloor \frac{y-y'}{r} \right\rfloor$, where (x', y') is the location of a sensor node. The depth of a zone is calculated to find its distance from a given zone. For a zone with *zone_ID* (a, b), its *depth*, $d_{zone} = max(|a_z - a|, |b_z - b|)$ with respect to a zone with *zone_ID* (a_z, b_z). In Figure 3.1, the value of d_{zone} of zone $(0, 3)$ is 1 from zones $(0, 2), (1, 2)$ and $(1, 3)$. Similarly, the value of d_{zone} of zone $(1, 2)$ is 1 from zones $(0, 1), (0, 2), (0, 3), (1, 1), (1, 3), (2, 1), (2, 2)$ and $(2, 3)$. A leader is elected in a zone called *zone leader* ZL, that collects and aggregates all the data coming from the local sensor nodes in that zone. All the remaining nodes are addressed as *non-ZLs*. ZL_status flag of a node indicates whether a node has been selected

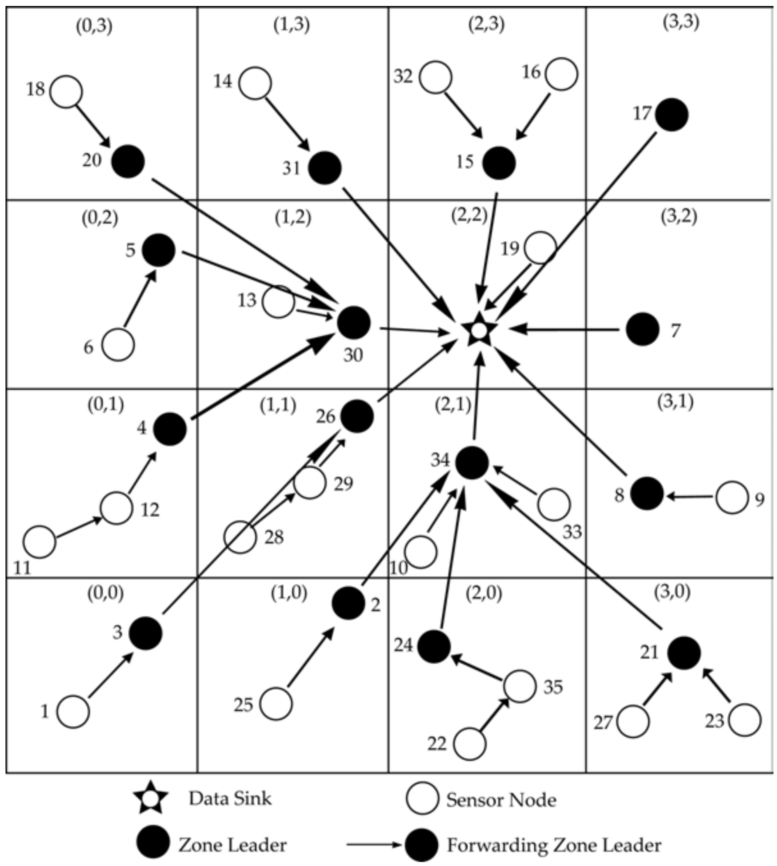

Fig. 3.1: A sample network divided in zones.

as a ZL or not. We assume that there is only one data sink which is located at the zone called *root zone* and the data sink acts as a *root ZL*. The distance of a zone (a, b) from the root zone (a_{rz}, b_{rz}), $d(a, b)$ is calculated by *Euclidean distance*, $d(a, b) = \sqrt{(a_{rz} - a)^2 + (b_{rz} - b)^2}$ [7]. Again considering Fig. 3.1, the value of $d(0, 3)$ for zone $(0, 3)$ is 2.24 and the value of $d(1, 2)$ for zone $(1, 2)$ is 1 from root zone $(2, 2)$. A zone may be further subdivided into clusters having *cluster head* CH, when ZL fails to cover all the local sensor nodes of that zone due to its buffer limitation. All the remaining nodes of a cluster are termed as *non-CH*s. For example, referring to Fig. 3.2, if the number of packets that a ZL can aggregate including its own packet i.e., buffer capacity $Buffer\_Cap$ of a ZL is 15 then the total number of nodes of that zone N_z which is 19, exceeds the buffer capacity of the ZL. Therefore, the ZL fails to accumulate the packets of all the nodes of that zone. In that case, clusters are formed inside that zone to cover all the nodes.

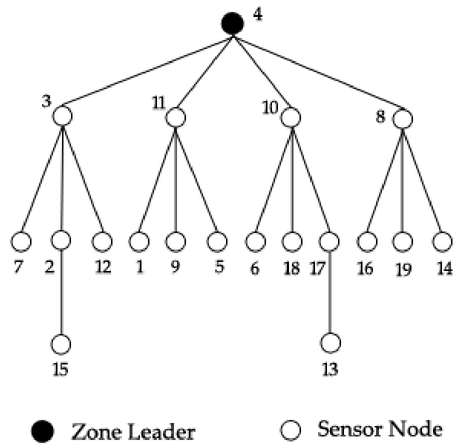

Fig. 3.2: Illustrating buffer limitation of the ZL in a zone.

3.2.2 Assumptions and Challenges

For sensor network, we make the following assumptions:

a) All nodes are homogeneous, static and have same energy supply initially.

b) A node is aware of its own location and the location of the data sink.

c) TDMA is used to avoid collision in intra-zone and inter-zone transmission/reception.

d) A sensor node turns the radio "on" only during message transmission/reception. Otherwise, the radio will be remained in "sleep mode".

There are some challenges involved in the network.

a) How to select a ZL in a zone such that energy consumption for data collection is minimized?

b) How to form clusters, if ZL fails to cover its joined nodes?

c) How to transmit data by all ZLs and CHs to the data sink?

d) How to perform ZL and CHs rotation in a zone in each round so that the network lifetime is increased?

3.3 Data Gathering Methodology

3.3.1 Zone Leader Selection

In a zone, a ZL sends a message to all the remaining sensor nodes of the zone to declare its leadership role. A ZL is selected through the cooperation of sensor nodes and maintained consistently in a zone for a particular round. The algorithm for the zone leader selection, ZL_SELECT is given in Algorithm 3.1. Initially, a sensor node is selected as a ZL in a zone as follows:

If there is only one sensor node in the zone then that node declares itself as the ZL. Otherwise, if the ZL_status flags of all the sensor nodes of that zone, are not set, then the node which is closer to the data sink announces its leadership role through a message with the ZL_status set. The reason for this heuristic is that if we select a node as ZL in a zone whose distance to data sink is less than any other nodes of that zone, the less energy will be consumed in forwarding the data to the sink. If more than one node in the zone set their ZL_status flag then the node bearing the highest $node\_ID$ is selected as ZL.

Algorithm 3.1 Zone leader selection algorithm: ZL_SELECT

Input: The total number of sensor nodes of a zone (a, b), N_z.
Output: ZL with ZL_status flag set.
Set ZL_status flag of each sensor node to 0;
if $N_z = 1$ **then**
    Set the ZL_status of that node as ZL;
else
    Set the ZL_status of a node as ZL which is the nearest to the data sink;
    if ZL_status of more than one node are set **then**
        Set the ZL_status of the node which has the highest $node\_id$;
    end if
end if

For example, referring to Fig. 3.1, node with $node\_id$ 3 is selected as a ZL in zone $(0, 0)$ as it is closer to the sink than the other local node with $node\_id$ 1.

3.3.2 Handling Multiple Clusters in a Zone

The nodes in the same zone may form multiple clusters, if ZL fails to accumulate data of all local non-ZL nodes due to its buffer limitation. If there are multiple clusters in a zone, a CH is elected in each cluster and the nodes within a cluster send their data to the local CH. Finally, all CHs send their aggregated data to the ZL or CHs of the nearest zone. Actually, all nodes send their data to ZL by maintaining a *Shortest Path Tree* [4]. The solution to the buffer limitation problem of ZL is to find the hop count limit M, within which nodes can send their data to ZL. Clusters are formed by estimating M. Referring to Fig. 3.3, we can find $Buffer\_Cap$ as follows.

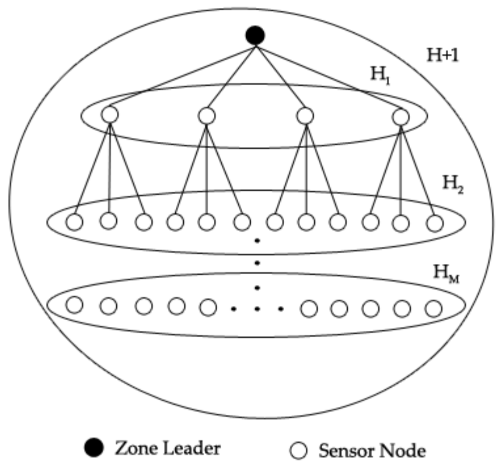

Fig. 3.3: Illustrating calculation of hop count, M in a zone.

$$Buffer\_Cap = H + 1 \tag{3.1}$$

where as,

$$H = H_1 + H_2 + H_3 + ... + H_{M-1} + H_M \tag{3.2}$$

and $H_1 = \delta$, $H_2 = H_1 \times (\delta - 1)$, $H_3 = H_2 \times (\delta - 1)$,..., $H_{M-1} = H_{M-2} \times (\delta - 1)$, $H_M = H_{M-1} \times (\delta - 1)$ and δ = average node degree. In Fig. 3.3, a tree with $\delta = 4$ is shown, where $H_1 = 4$, $H_2 = 4 \times (4 - 1) = 12$ and so on. Referring Equations

3.1 and 3.2, we can write,

$$\begin{aligned}
Buffer_Cap &= \delta + \delta(\delta-1) + \delta(\delta-1)^2 + ... + \delta(\delta-1)^{M-2} + \delta(\delta-1)^{M-1} \\
&\quad +1 \\
&= \delta\left\{1 + (\delta-1) + (\delta-1)^2 + ... + (\delta-1)^{M-2} + (\delta-1)^{M-1}\right\} \\
&\quad +1 \\
&= \delta\left\{\frac{(\delta-1)^M - 1}{(\delta-1)-1}\right\} + 1 \\
&= \delta\left\{\frac{(\delta-1)^M - 1}{\delta-2}\right\} + 1 \qquad (3.3)
\end{aligned}$$

From Equation 3.3, we can find,

$$M = \left\lfloor \log_{(\delta-1)}\left\{\frac{(Buffer\_Cap-1)(\delta-2)}{\delta} + 1\right\}\right\rfloor \qquad (3.4)$$

Thus, a non-ZL node must send data to ZL within-
$\left\lfloor \log_{(\delta-1)}\left\{\frac{(Buffer\_Cap-1)(\delta-2)}{\delta} + 1\right\}\right\rfloor$ hops. Otherwise, clusters are formed inside the zone using the value of M as described in Algorithm 3.2, titled CF_ZONE and an example is illustrated in Fig. 3.4. One advantage of maintaining CHs over ZL is that it provides solution of the buffer limitation problem. Moreover, aggregation load is reduced in each sensor node and fixed energy efficiency is possible to obtain in zone wide. We take $Buffer\_Cap$ as user-defined parameter. For example, if $Buffer\_Cap = 17$ and $\delta = 4$ then considering Equation 3.4, we get $M = 2$. Similarly, if $Buffer\_Cap = 53$ and $\delta = 4$, then M is 3.

Let us explain how the cluster formation algorithm, CF_ZONE works in a zone (a, b). Initially, the shortest path tree T is constructed (having ZL as root) in the zone that covers all sensors of that zone. We apply *Dijkstra's Algorithm* [4] to build the T. The cluster formation algorithm CF_ZONE repeatedly finds CHs on the T. Let z is the farthest leaf node rooted at ZL on the T. There may be two cases described as follows.

- **Case 1:** The first case is that z does not become a CH yet. In this situation, the T is traversed from z towards the ZL to search its M-hop parent (M is calculated using Equation 3.4). Suppose, y is the M-hop parent of z. As z is the farthest node, all other descendants of y are able to reach y within M hops. Therefore, we can mark y as a CH as it is the nearest one to the ZL that can connect the sensors in the fringe of the zone. After then, the T is updated

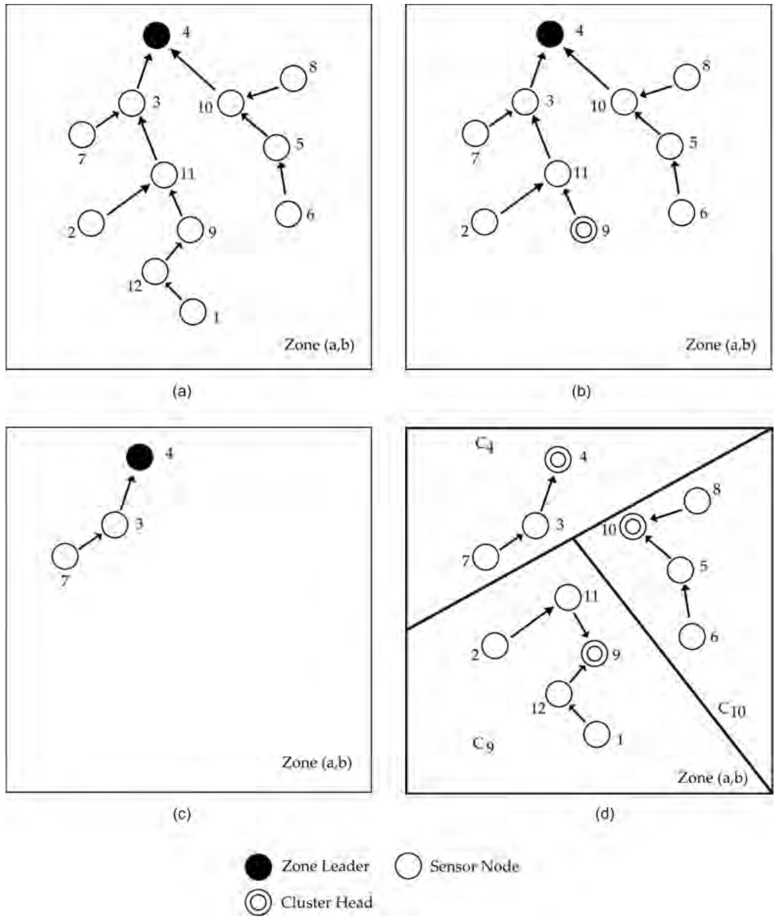

Fig. 3.4: Illustrating cluster formation when $M = 2$: (a) initial configuration, (b) after first iteration, (c) after second iteration, and (d) the final configuration.

by pruning all the descendants of y. The pruned sensors are joined with y in the cluster, C_y.

It is noted that y is still kept on the T in order to join more sensors with y, if possible, in future iterations. When ZL is reached during the iteration of searching the M-hop parent of z, the algorithm exits because all the nodes on the updated T are within M hops to the ZL and the ZL is redefined as a CH.

- **Case 2:** The second and the final case is that z has already become a CH. In this situation, we try to join more sensors with z, if possible, in order to decrease the total number of CHs. To obtain more sensors in cluster C_z, we search x which is $\lfloor \frac{M}{2} \rfloor$-hop parent of z. As z is the farthest leaf node on the current T, all other descendants of x must be within $\lfloor \frac{M}{2} \rfloor$ hops away from x so that they are able to reach z within M hops. Therefore, x and its all other descendants are joined in C_z.

Algorithm 3.2 Cluster formation algorithm: CF_ZONE

```
Input: T rooted at ZL, Buffer_Cap and δ.
Output: Clusters C_n s where CH is the sensor with node_id n.
M := ⌊ log_(δ−1) { (Buffer_Cap−1)(δ−2)/δ + 1 } ⌋;
while T ≠ ∅ do
    Calculate the farthest leaf z of T;
    if z ≠ CH then
        for j := 1 → M do
            y := Parent(z);
            z := y;
            if y = ZL then
                break;
            end if
        end for
        if y ≠ ZL then
            T := T − {Descendants of y};
            C_y := C_y ∪ {Descendants of y};
        else
            C_y := C_y ∪ {All nodes of T};
            T := ∅;
        end if
    else
        if M = 1 then
            T := T − {z};
        else
            for j := 1 → ⌊M/2⌋ do
                x := parent(z);
                z := x;
                if x = ZL then
                    break;
                end if
            end for
            if x ≠ ZL then
                T := T − {x} − {Descendants of x};
                C_z := C_z ∪ {x} ∪ {Descendants of x};
            else
                C_z := C_z ∪ {Remaining non − CHs of T};
                T := ∅;
            end if
        end if
    end if
end while
```

where $M := \left\lfloor \log_{(\delta-1)} \left\{ \frac{(Buffer\_Cap-1)(\delta-2)}{\delta} + 1 \right\} \right\rfloor$.

To clarify the algorithm CF_ZONE, we give a detail example in Fig. 3.4 where 12 sensors are spread in a zone (a, b) and suppose M is found to 2 using Equation 3.4, which means that it is required for each sensor to forward its data to the ZL within two hops. The constructed T among the sensors rooted at ZL with *node_id* 4 is depicted in Fig. 3.4(a). In the first iteration, sensor with *node_id* 1 is found as the farthest leaf vertex on the T with 5 hops away from the ZL, i.e., $z = 1$. Its 2-hop parent y on current T is sensor with *node_id* 9, which is marked as a CH. All the descendants of y i.e., sensors with *node_id*s 1 and 12 are eliminated from the T. Sensor with *node_id* 9 is still kept on the updated tree T. The result is depicted in Fig. 3.4(b). In the second iteration, the farthest leaf z on the updated T turns to be sensors with *node_id*s 2, 6 and 9 with 3 hops away from the ZL. Node with *node_id* 9 is already taken as CH so we try to join more sensors with it. The parent x of node with *node_id* 9 i.e, node with *node_id* 11 is in $\lfloor \frac{M}{2} \rfloor = 1$-hop away. Therefore, parent with *node_id* 11 and sibling with *node_id* 2 are also joined with node with *node_id* 9. In a similar way, node with *node_id* 10 acts as a CH. Fig. 3.4(c) shows the operation. Now, the farthest leaf z is node with *node_id* 7. Its parent ZL acts as a CH. Fig. 3.4(d) gives the final result showing clusters C_4, C_9 and C_{10} which have been numbered according to the CHs. CHs of C_4, C_9 and C_{10} are nodes with *node_id*s 4, 9 and 10 respectively.

3.4 Overview of ZDG

3.4.1 Details of ZDG

After dividing the network terrain into zones, ZLs are elected through zone leader selection strategy. If all non-ZLs are covered by ZL, ZLs allocate schedule to non-ZLs to receive packets. After collecting packets, ZL transmits the aggregated data to ZL or CH of nearest zone which is closer to root zone. On the contrary, if all nodes are not covered then clusters are created inside in zones by Algorithm 3.2, CF_ZONE. All CHs send their aggregated data to ZL or CH of the nearest zone that is closer to the root zone. The next round follows the similar steps. The overall system architecture is shown in Figure 3.5.

Fig. 3.5: The details of zone based data gathering method ZDG.

3.4.2 Data Gathering and Sending to Sink

A ZL or CH sends the aggregated data to a ZL or CH of the nearest zone which is closer to the root zone to gather the packets by data sink. Referring to Fig. 3.1, for example, ZL with *node_id* 3 wishes to send aggregated packets to data sink/root ZL. The transmission is done by the following procedure: Zone $(0,0)$ has outer zones $(0,1),(1,0)$ and $(1,1)$, whose d_{zone} value is 1. d_{zone} for zone $(0,1)$ with respect to zone $(0,0) = max(|0-0|, |0-1|) = 1$. Similarly, the value of d_{zone} for zones $(1,0)$ and $(1,1)$ with respect to zone $(0,0)$ is 1. Therefore, ZL with *node_id* 3 of zone $(0,0)$ forwards its aggregated packet to one of these zones which is nearest to data sink/root ZL. Now, the value of $d(0,1)$ for zone $(0,1)$ from root zone $(2,2) = \sqrt{(2-0)^2+(2-1)^2} = 2.24$. Similarly, the values of $d(a,b)$ for zones $(1,0)$ and $(1,1)$ are 2.24 and 1.41 respectively. Therefore, ZL

with *node_id* 3 transmits its aggregated packet to ZL with *node_id* 26 of zone $(1,1)$. By following the same procedure, ZL with *node_id* 26 forwards the packet data sink/root ZL of root zone.

It is noted that when multiple clusters are formed in zones, if a CH of one zone sends data to the nearest zone consisting of multiple clusters then all CHs of the nearest zone generate reply packet individually and start reply timer simultaneously with an expiration time inversely proportional to their individual residual energy. The first CH that has an expired timer transmits the reply packet back to the requester CH, while the remaining CHs immediately stop their timers. Fig. 3.6 depicts a scenario of multiple clusters per zone. For example, if CH

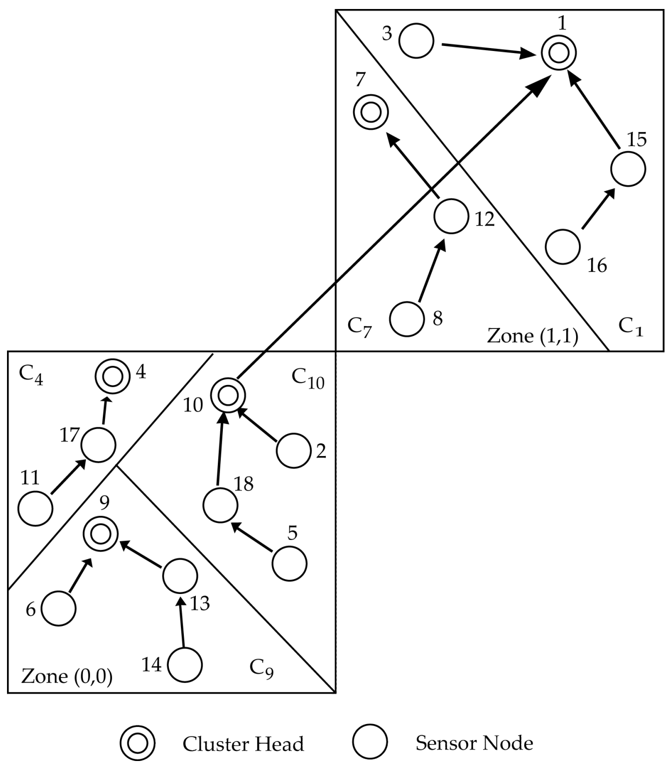

Fig. 3.6: Scenario of data transmission between multiple clustered zones.

with *node_id* 10 of zone $(0,0)$ wants to forward its data to the CH of the nearest zone $(1,1)$ then CHs with *node_id* 1 and 7 start their reply timers. Let CH with *node_id* 1 has more energy than CH with *node_id* 7. In this situation, timer of CH with *node_id* 1 expires before timer of CH with *node_id* 7. Therefore, CH with *node_id* 1 receives the data of CH with *node_id* 10 and CH with *node_id* 7 cancels its timer.

ZL selection and CH selection in a zone are done in each round. A node is selected as ZL, if it has remaining energy greater than average residual energy of all nodes of that zone, and is the nearest to data sink/root ZL than any other nodes.

3.4.3 Energy Efficiency and Deadlock Avoidance

Our method ZDG provides the following facilities:

This scheme balances load and reduces energy consumption and thus, increases network lifetime because only the selected sensors act as ZLs and CHs and send data to the sink. No deadlock situation arises in ZDG. For example,

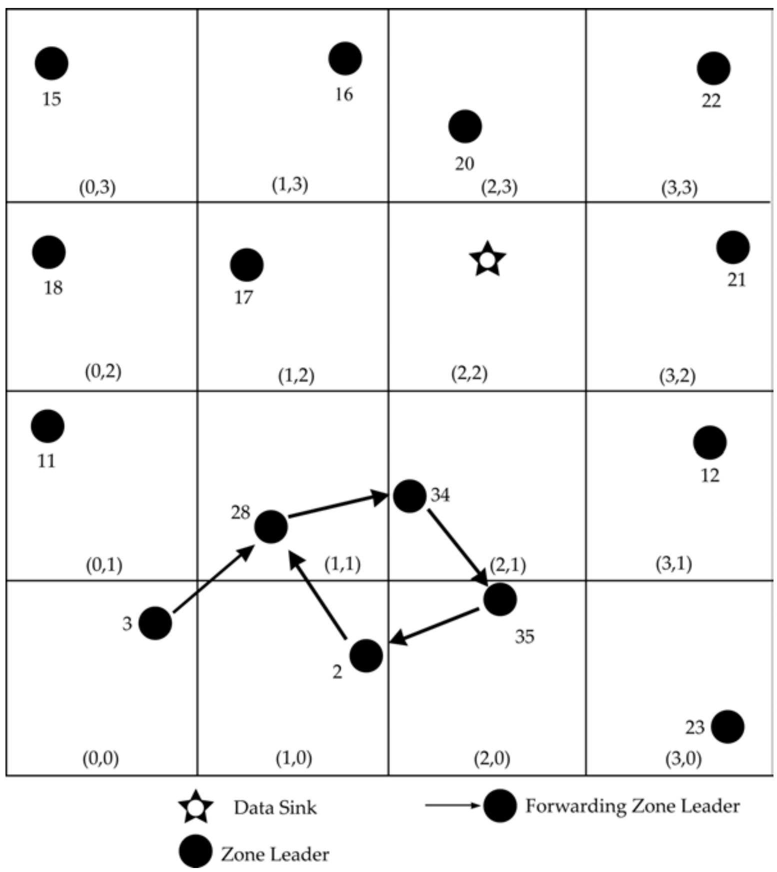

Fig. 3.7: A scenario of deadlock situation.

considering Figure 3.7, if the nearest ZL selection strategy is followed then ZL with *node_id* 3 sends data to the nearest ZL with *node_id* 28. Then, ZL with *node_id* 28 transfers it to nearest ZL with *node_id* 34, which in turns forwards the data to the nearest ZL with *node_id* 35. After then, ZL with *node_id* 35 transfers the data to ZL with *node_id* 2 which sends the data again to ZL with *node_id* 28. Thus, a cycle or deadlock situation arises.

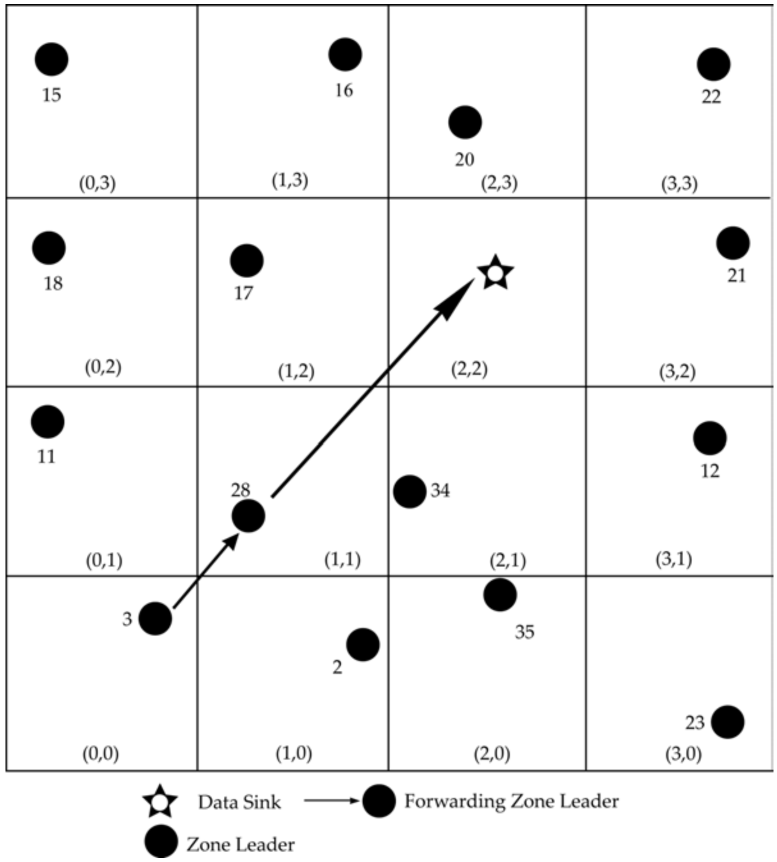

Fig. 3.8: Remedy of deadlock situation.

To overcome from this deadlock situation, we use the nearest zone selection strategy instead of the nearest ZL selection method. In this strategy, ZL with *node_id* 3 sends its gathered data to ZL with *node_id* 28 of the nearest zone $(1, 1)$ and finally, ZL with *node_id* 28 concludes the transmission by sending it to ZL of the nearest zone i.e., sink of root zone. Figure 3.8 depicts the remedy of the deadlock situation.

Chapter 4

Analysis of ZDG

4.1 Overview

In this chapter, the value of energy consumption of a zone in a data gathering round i, $E_{DGR}(i)$ is demonstrated. In addition with discussing individual parts of $E_{DGR}(i)$ separately, time complexities are analyzed.

4.2 Energy Consumption by a Sensor

The respective ways of calculating energy dissipation for each of the selected simulation methods are discussed in this section. As shown in Fig. 4.1 [8–10, 24], there are three modules contributing to the energy consumption: the Radio module, Sensor module, and MCU module.

• **Radio model:** The *radio module* is responsible for wireless communication among nodes. A typical radio module used in wireless devices is shown in Fig. 4.1 [8–10, 24]. The "Transmit Electronics" represents electronics circuit performing signal modulation. The "Tx Amplifier" is used to amplify the modulated signal and output it to the antenna. The "Receive Electronics" is used to decode the modulated signal. E_{elec} is the energy needed for modulating or demodulating one bit of the circuits. ϵ_{amp} is the energy for the amplifier circuit to transmit one bit to an area of radius $d = 1$ meter (i.e., πd^2). In a real device, the *transmit module* ("Transmit Electronics" and "Tx Amplifier") normally stays in "sleep mode". It only wakes up when there is any bit that needs to be sent. The *receiver module* ("Receive Electronics") performs the reverse function. It needs to be "on" to receive messages.

By considering the popular *Heinzelman's first order radio model* as shown in

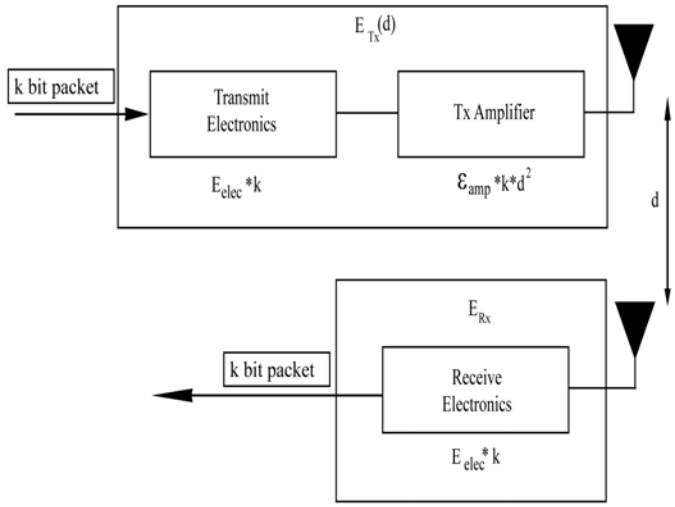

Fig. 4.1: Heinzelman's first order radio model [8–10, 24].

Fig. 4.1, transmission and receiving costs are given by:

$$E_{Tx}(k,d) = \begin{cases} kE_{elec} + k\epsilon_{fs}d^2\ , \text{ if } d < d_0 \\ kE_{elec} + k\epsilon_{mp}d^4, \text{ if } d \geq d_0 \end{cases} \tag{4.1}$$

$$E_{elec} = e_t + e_r \tag{4.2}$$

$$d_0 = \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}} \tag{4.3}$$

$$E_{Rx}(k) = kE_{elec} \tag{4.4}$$

$E_{Tx}(k,d)$ represents the energy needed to spread k bits to an area of radius d, while $E_{Rx}(k)$ is the energy needed to de-modulate k bits. Both the free space ($\epsilon_{fs}d^2$ power loss) and the multi-path fading ($\epsilon_{mp}d^4$ power loss) channel model are used as considered in [8–10, 24]. d is the distance between transmitter and receiver. The baseline energy consumption levels at the transmitter and receiver radios are indicated by e_t and e_r respectively. d_0 is called as *threshold distance.*

- **Sensor board, MCU (CPU board, Memory board), and Radio board of a sensor network:** These boards work in two modes: "full action" and "sleep". In the "sleep mode", the energy dissipation is almost zero. The "full action" mode consumes energy as shown in Fig. 4.2 [6].

SYSTEM SPECIFICATIONS		
Currents		**Example Duty Cycle**
Processor		
Current (full operation)	8 mA	1
Current sleep	8 µA	99
Radio		
Current in receive	8 mA	0.75
Current transmit	12 mA	0.25
Current sleep	2 µA	99
Logger Memory		
Write	15 mA	0
Read	4 mA	0
Sleep	2 µA	100
Sensor Board		
Current (full operation)	5 mA	1
Current sleep	5 µA	99

Fig. 4.2: Current of boards in sensor node MICA2DOT (MPR 500) [6].

4.3 Energy Consumption Calculation for ZDG

There may be two cases:

- **Case 1:** No cluster formation i.e., ZL can accumulate data of all non-ZLs.
- **Case 2:** Cluster formation i.e., when ZL of a zone fails to accumulate data of all non-ZLs.

Now, we calculate the energy consumption for each zone in i-th data gathering round.

4.3.1 Energy Consumption for Zone Formation

Denoting the control packet by k_0, the total energy consumption for a zone creation with a ZL during i-th data gathering round (DGR) is calculated as follows.

Case 1: No Cluster Formation

$$E_{zf}(i) = k_0 \sum_{p,j \leq N_z} (E_{elec} + \epsilon_{fs} d_{p,j}^2) + k_0 \sum_{p \leq N_z - 1} (E_{elec} + \epsilon_{fs} d_{ZL,p}^2) + k_0(e_t + \epsilon_{fs} d_{ZL,p}^2) + k_0(N_z - 1)e_r \quad (4.5)$$

This equation can be briefly explained as follows:

The first term of Equation 4.5 denotes the selection of a node to act as ZL where, $d_{p,j}$ is the distance between nodes p and j. This is the summation of energy consumptions by all nodes, N_z to send their information to each other.

After selection of ZL, the ZL announces its leadership role to all the remaining nodes $(N_z - 1)$. The second term stands for this event. Here, $d_{ZL,p}$ is the distance between ZL and non-ZL node, p. The third term represents the transmission of time schedule by ZL. Finally, the remaining term is for all non-ZLs to receive their time schedule from ZL.

Case 2: Cluster Formation

$$
\begin{aligned}
E_{zf}(i) \quad = \quad & k_0 \sum_{p,j \leq N_z} (E_{elec} + \epsilon_{fs} d_{p,j}^2) + k_0 \sum_{j \leq N_z - Q_t} (E_{elec} \\
& + \epsilon_{fs} d_{p,j}^2) + k_0 \sum_{p \leq Q_t} (E_{elec} + \epsilon_{fs} d_{j,p}^2) \\
& + k_0 \sum_{p \leq Q_t} (e_t + \epsilon_{fs} d_{p,j}^2) + k_0 (N_z - Q_t) e_r \qquad (4.6)
\end{aligned}
$$

The first term of Equation 4.6 is for selection of nodes to be CHs. This is the addition of energy consumptions by all nodes, N_z to send their information to each other. When a node becomes CH, it announces its role and is heard by only non-CH nodes $(N_z - Q_t)$. The second term indicates these events. $d_{p,j}$ is the distance between CH p and non-CH node j, and Q_t indicates the number of CHs. Each non-CH node needs to send a control packet to join a CH. The third term is the energy consumption for processing all non-CH nodes to CHs . The fourth term represents the transmission of time schedule by CHs. Finally, the remaining term is for all non-CHs to receive their time schedule from CHs.

4.3.2 Energy Consumption for Sending Data

Basically, a ZL or CH requests to send its data to its neighboring ZL or CHs of zone which is closer to the sink. Considering all data packets emerging from the neighbor zones as a load on a zone, the number of such packets l can be calculated by $\sum_{j \leq l} N'_j$. $d_{z,z'}$ is the distance of adjacent square zones, z and z'. The energy consumption for sending data during a DGR is given by-

Case 1: No Cluster Formation

$$
E_{sd}(i) \quad = \quad 2k_0 (E_{elec} + \epsilon_{mp} d_{z,z'}^4) \sum_{j \leq l} N'_j \qquad (4.7)
$$

The term of Equation 4.7 demonstrates the reception of packets by ZL coming from its depth-1 zone, the acknowledgement sent by ZL and further sending. It is twice because the same energy is consumed for further sending initiated by the ZL towards the data sink as needed for reception and acknowledgement.

Case 2: Cluster Formation

$$\begin{aligned} E_{sd}(i) &= k_0(Q_t e_r + e_t + \epsilon_{mp} d^4_{z,z'}) \sum_{j \le l} N'_j + k_0(E_{elec} \\ &\quad + \epsilon_{mp} d^4_{z,z'}) \sum_{j \le l} N'_j \end{aligned} \tag{4.8}$$

The first term of Equation 4.8 demonstrates the reception of packets by all CHs Q_t, coming from its depth-1 zone and the acknowledgement sent by one CH. The second term is for further sending initiated by the CH towards the data sink.

4.3.3 Energy Consumption for Inter-zone Communication

The total number of packets transmitted by a zone is the summation of the generated packets by ZL or CHs and forwarding packets. Denoting k be the average packet length, $d_{g,r}$ be the distance between ZL or CH of adjacent zones and n_t be the total packets of a zone, the energy consumption between neighboring zones during a round is as follows.

Case 1: No Cluster Formation

$$E_{inter}(i) = kn_t(e_t + \epsilon_{mp} d^4_{g,r}) + ke_r(n_t - 1) \tag{4.9}$$

where, the first term stands for the energy consumption by the ZL to transmit all the packets (including its own packet)n_t to the outer zone and the second term is for the energy consumption by the ZL to receive all the incoming packets $(n_t - 1)$ from the outer zones.

Case 2: Cluster Formation

$$E_{inter}(i) = kn_t(e_t + \epsilon_{mp} d^4_{g,r}) + ke_r(n_t - Q_t) \tag{4.10}$$

where, the first term stands for the energy consumption by CHs to transmit all the packets (including their own packets) n_t to the outer zone and the second term is for the energy consumption by CHs to receive all the incoming packets $(n_t - Q_t)$ from the outer zones.

4.3.4 Energy Consumption for Intra-zone Communication

During a DGR, each sensor encapsulates its observed information in a data packet and then transmit this to the corresponding ZL or CH. The ZL or CH accumulates all observation packets to aggregate the observations of area of the zone. By considering both transmission and reception events, we obtain total energy dissipation for communication between nodes in a zone as follows.

Case 1: No Cluster Formation

$$E_{intra}(i) \;=\; k\left(\sum_{j\leq N_z-1}(e_t+\epsilon_{fs}d_{j,ZL}^2)\right)+ke_r(N_z-1) \qquad (4.11)$$

The first term of Equation 4.11 is the total energy consumption for transmission of data by all non-ZL nodes $(N_z - 1)$ and the final term is used for reception of data packets by ZL from $(N_z - 1)$ number of non-ZL nodes.

Case 2: Cluster Formation

$$E_{intra}(i) \;=\; k\left(\sum_{j\leq N_z-Q_t}(e_t+\epsilon_{fs}d_{j,p}^2)\right)+ke_r(N_z-Q_t) \qquad (4.12)$$

The first term of Equation 4.12 is the total energy consumption for transmission of data by all non-CH nodes $(N_z - Q_t)$ and the last term is used for reception of data packets by CHs from $(N_z - Q_t)$ number of non-CH nodes.

4.3.5 Energy Consumption for Data Aggregation

The energy consumption for data aggregation (whether cluster is formed or not) can be calculated as follows.

$$E_{agg}(i) = kN_zE_b \qquad (4.13)$$

The energy consumption for data aggregation given in Equation 4.13 is the total number of nodes in a zone multiplied by the number of bits in a packet and energy required to aggregate one bit of data E_b.

4.3.6 Energy Consumption in Data Communications per Round

The total energy for data communication per round is as follows.

$$E_{comm}(i) = E_{inter}(i) + E_{intra}(i) + E_{agg}(i)$$

This means that the total energy consumed for data communications in each round is the summation of energy consumptions for inter-zone communication, intra-zone communication and data aggregation.

Case 1: No Cluster Formation

By Equations 4.9, 4.11 and 4.13, we have,

$$\begin{aligned} E_{comm}(i) &= kn_t(e_t + \epsilon_{mp} d_{g,r}^4) + ke_r(n_t - 1) \\ &\quad + k\left(\sum_{j \leq N_z - 1} (e_t + \epsilon_{fs} d_{j,ZL}^2)\right) + ke_r(N_z - 1) \\ &\quad + kN_z E_b \end{aligned} \tag{4.14}$$

In the case of "no cluster formation", the first two terms of Equation 4.14 are the estimated energy consumptions regarding the inter-zone communication. The next two terms represent the calculated energy consumptions for intra-zone communication, and the final term is the energy consumption for data aggregation.

Case 2: Cluster Formation

By Equations 4.10, 4.12 and 4.13, we formulate,

$$\begin{aligned} E_{comm}(i) &= kn_t(e_t + \epsilon_{mp} d_{g,r}^4) + ke_r(n_t - Q_t) \\ &\quad + k\left(\sum_{j \leq N_z - Q_t} (e_t + \epsilon_{fs} d_{j,p}^2)\right) + ke_r(N_z - Q_t) \\ &\quad + kN_z E_b \end{aligned} \tag{4.15}$$

In the case of "cluster formation", the first two terms of Equation 4.15 are the estimated energy consumptions regarding the inter-zone communication. The next two terms represent the calculated energy consumptions for the intra-zone communication, and the last term is the energy consumption for data aggregation.

4.3.7 Total Energy Consumption per Round in a Zone

We find total energy consumption in a zone in i-th round as follows.

$$E_{DGR}(i) = E_{zf}(i) + E_{sd}(i) + E_{comm}(i)$$

The above equation demonstrates that the total energy consumption in a data gathering round (DGR) is the summation of energy consumptions for zone formation, for sending data, and for data communication.

Case 1: No Cluster Formation

Using Equations 4.5, 4.7 and 4.14, we get,

$$\begin{aligned} E_{DGR}(i) = {} & k_0 \sum_{p,j \le N_z} (E_{elec} + \epsilon_{fs} d_{i,j}^2) + k_0 \sum_{p \le N_z - 1} (E_{elec} \\ & + \epsilon_{fs} d_{ZL,p}^2) + k_0(e_t + \epsilon_{fs} d_{ZL,p}^2) + k_0(N_z - 1)e_r \\ & + 2k_0(E_{elec} + \epsilon_{mp} d_{z,z'}^4) \sum_{j \le l} N_j' + k n_t (e_t \\ & + \epsilon_{mp} d_{g,r}^4) + k e_r (n_t - 1) + k \left(\sum_{j \le N_z - 1} (e_t + \epsilon_{fs} d_{j,ZL}^2) \right) \\ & + k e_r (N_z - 1) + k N_z E_b \end{aligned} \quad (4.16)$$

In the case of "no cluster formation", the first four terms of Equation 4.16 are the estimated energy consumptions regarding zone formation. The next term represents the calculated energy consumption for data sending, and rest of the terms are the energy consumptions for data communications.

Case 2: Cluster Formation

Using Equations 4.6, 4.8 and 4.15, we have,

$$\begin{aligned}
E_{DGR}(i) &= k_0 \sum_{p,j \leq N_z} (E_{elec} + \epsilon_{fs} d_{p,j}^2) + k_0 \sum_{j \leq N_z - Q_t} (E_{elec} \\
&\quad + \epsilon_{fs} d_{p,j}^2) + k_0 \sum_{p \leq Q_t} (E_{elec} + \epsilon_{fs} d_{j,p}^2) \\
&\quad + k_0 \sum_{p \leq Q_t} (e_t + \epsilon_{fs} d_{p,j}^2) + k_0 (N_z - Q_t) e_r \\
&\quad + k_0 (Q_t e_r + e_t + \epsilon_{mp} d_{z,z'}^4) \sum_{j \leq l} N_j' + k_0 (E_{elec} \\
&\quad + \epsilon_{mp} d_{z,z'}^4) \sum_{j \leq l} N_j' + k n_t (e_t + \epsilon_{mp} d_{g,r}^4) \\
&\quad + k e_r (n_t - Q_t) + k \left(\sum_{j \leq N_z - Q_t} (e_t + \epsilon_{fs} d_{j,p}^2) \right) \\
&\quad + k e_r (N_z - Q_t) + k N_z E_b
\end{aligned} \tag{4.17}$$

In the case of "cluster formation", the first five terms of Equation 4.17 are the estimated energy consumptions regarding the zone formation. The next two terms represent the calculated energy consumption for data sending and the rest of the terms are the energy consumptions for data communications.

4.4 Cost Analysis of ZDG

In this section, the time complexity calculation of ZDG is discussed.

The zone is calculated by each sensor node based on its position and the reference location of a sensor node. The ZL information is distributed with a ZL_status flag inserted in the messages. Therefore, the cost for building zone is $O(1)$.

There are N_z sensors in a zone, and $O(N_z^2)$ is needed to make the shortest path tree using Dijkstra's algorithm [4]. It takes N_z steps to find the farthest leaf of that tree, and M repetitions to find the CH. Therefore, $O(N_z^2 + N_z M)$ time is required for finding CHs and joining non-CHs. As a result, $O(N_z^2) + O(N_z^2 + N_z M)$ time is required for the cluster formation in a zone.

Chapter 5

Experimental Results

5.1 Overview

The acceptance of ZDG depends on the comparative performance analysis and superiority compared with the existing schemes. This chapter presents the detail experimental setup of the ZDG and the existing schemes. The experiments are carried out using the popular simulator OMNeT++ which can be used for analyzing all aspects of WSNs ranging from routing to clustering to communications. For investigating the performance, each of the components anticipated in the ZDG scheme is studied separately with the latest and popular schemes EBDG [35] and Multihop-LEACH [32].

5.2 Simulation Environment

5.2.1 Simulation Platform

The simulation of ZDG over WSN is run on 2.53 *GHz Intel Core i5* processor with 4 *GB* memory. The operating system used to run simulation is *Windows 7*.

We use OMNeT++ [19] as a simulation tool. OMNeT++ is a public-source, component-based, modular and open-architecture simulation environment with strong GUI support and an embeddable simulation kernel. Its primary application area is the simulation of communication networks, IT systems, querying networks, hardware architectures, or even business processes.

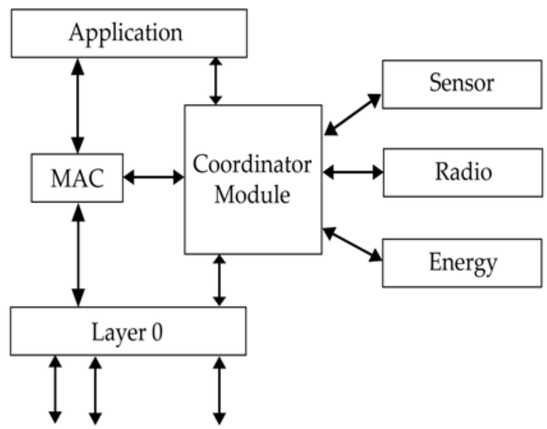

Fig. 5.1: A simulated sensor node structure [23].

5.2.2 Sensor Node Architecture in OMNeT++

The Sensor Network Research Group at Louisiana State University has defined a generic sensor node. Based on this generic design, a simulated sensor node has been built as illustrated in Fig. 5.1 [23].

• The *Layer 0 module* represents the physical layer of a sensor node. It is responsible for making connections between the node and its neighbors, and forwarding messages from a higher layer to its neighbors, and vice versa.

• The *MAC module* represents pre-processing packet layers. It consists of gates (in/out) and queues (incoming queue and outgoing queue). When the queue is full, it deletes some of the oldest messages to make room in the queue for new messages. It helps to evaluate performance of the node.

• The *Application module* represents the application layer. Note that, each time after sending a message, the module automatically sends a *DECREASE_ENERGY* message to the *energy module* (through the *coordinator module*) to let the module decrease the energy by a number of energy units.

• The *Coordinator module* is an interface to connect all modules together. It categorizes an incoming message in order to deliver it to the right module. For example, when receiving a *DECREASE_ENERGY* message, it forwards the message to the *energy module*.

• The *Sensor module* represents the sensor board in a sensor node. If the *SENSOR_SWITCH* parameter is $ON(=1)$, the module consumes energy, therefore, after an interval (timer), the module sends a *DECREASE_ENERGY* message to the *energy module* (through the *coordinator module*). When the timer ticks, the waiting timer decreases. The waiting timer is set by SENSOR_REFRESH messages from the *application module*. If the waiting timer is zero, the module will turn off ($SENSOR\_SWITCH$ parameter is set to 0).

• The *Radio module* represents the radio board in a sensor node. If $RADIO\_SWITCH$ parameter is $ON(=1)$, the module consumes energy, therefore, after an interval (timer), the module sends a $DECREASE\_ENERGY$ message to the *energy module* (through the *coordinator module*).

• The *Energy module* represents the battery in a sensor node. If the module receives a $DECREASE\_ENERGY$ message, it decreases the energy level by a number of energy units.

• **Parameters of the sensor node in the simulation:**

∘ $CNNCTVTY$: Maximum connections a node has.
∘ $OCCUPATION$: Task of the node.
∘ PX: Position by *X*.
∘ PY: Position by *Y*.
∘ ID: *ID* of node.
∘ $FATHER\_ID$: *ID* of node for forwarding messages.
∘ $COMMUNICATION\_RADIUS$: Radius of the effective zone within which the node can communicate.
∘ $ENERGY$: Energy level.
∘ $SENSOR\_SWITCH$: Turn *ON*/*OFF* the sensor module.
∘ $RECEIVER\_SWITCH$: Turn *ON*/*OFF* the radio receiver module.
∘ $POWER\_SWITCH$: Turn *ON*/*OFF* the node.

• **Parameters for setting up the sensor network in the simulation:**

∘ $ZONE\_X$: Area by x.
∘ $ZONE\_Y$: Area by y.
∘ $FILE\_PATHS$: Paths to input/output files for the simulation.
∘ $NNODES$: Number of sensor node in the network.
∘ $NODE\_COMMUNICATION\_RADIUS$: Communication radius of each node.

5.3 Selected Methods and Types of Messages for Simulation

In evaluating the performance of ZDG, two popular methods are selected for comparisons: EBDG [35] and Multihop-LEACH [32]. There are 4 types of messages used in our simulation, each classified as either an external or an internal message:

• *SENSING_DATA* (external message): When a node receives this kind of messages, it forwards the message to its father node.

• *DECREASE_ENERGY* (internal message): When an energy module receives the message, it decreases the energy level of the node by the number of energy units contained in the message.

• *SENSOR_REFRESH* (internal message): When the sensor module gets the message, it resets the waiting timer.

• *TIMER* (internal message): It is a triggering message.

5.4 Simulating a Sensor Node

A sensor network includes an array of sensor nodes. To simulate such a sensor network in the simulation environment, we need a module called *manager* to help simulate tasks such as making connections among the nodes and saving the simulation results. To construct the sensor network, the manager module starts by first reading data in the *network.txt* file, which stores network configuration information, including sensor node positions, tasks, and sending. Then, it makes connections among nodes (it checks the coverage zone of all nodes to see if any node is in the coverage and makes connections between the node and the covered nodes). Finally, it controls the power switch (*POWER_SWITCH* parameter) of all nodes in the network.

5.5 Simulation Metrics

Overall, there are three types of metrics that are considered when comparing the performance: the total energy consumption, network lifetime and throughput in a fixed duty cycle.

• The *total energy consumption* is defined as the total energy that the total nodes of the network consume.

• The *network lifetime* is defined as the time before the first node dies.

• The *throughput* is the number of packets received by sink.

5.6 Simulation Results

5.6.1 Building Scenario

The simulation area is $640m \times 640m$ and the number of nodes in the WSN considered are 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800, 850, 900, 950 and 1000. We set $R = 60m$. The communication energy parameters are set as- $E_{elec} = 50\ nJ/bit$, $\epsilon_{fs} = 10\ pJ/bit/m^2$ and $\epsilon_{mp} = 0.0013pJ/bit/m^4$. We consider all nodes are of same initial energy $2J$, all control and data packets are of length 64 *bits* and 2000 *bits* respectively. Table 5.1 summarize the system parameters. The values of $Buffer\_Cap$ are 17 and 53. Simulation time has been set to random time 9000 *sec*.

Table 5.1: **Simulation parameters**

Parameters	**Value**
Node initial Energy	2 J
Data Packet Size	2000 bits
Signal Packet Size	64 bits
E_{elec}	50 nJ/bit
ϵ_{fs}	10 pJ/bit/m^2
ϵ_{mp}	0.0013 pJ/bit/m^4
Buffer_Cap	17, 53
Simulation Time	9000 sec

5.6.2 Performance Evaluation

Figures 5.2, 5.3 and 5.4 summarize the simulation results of ZDG, EBDG and Multihop-LEACH for $Buffer\_Cap = 17$.

The Fig. 5.2 is the summary of the energy consumption. In Fig. 5.2, ZDG's energy consumption delivers better results than the others in all cases. ZDG consumes less energy than EBDG and Multihop-LEACH because a few number of nodes are selected since ZL and CHs and gathered data by the selected nodes are transferred via the nearest zones. The result indicates that ZDG consumes

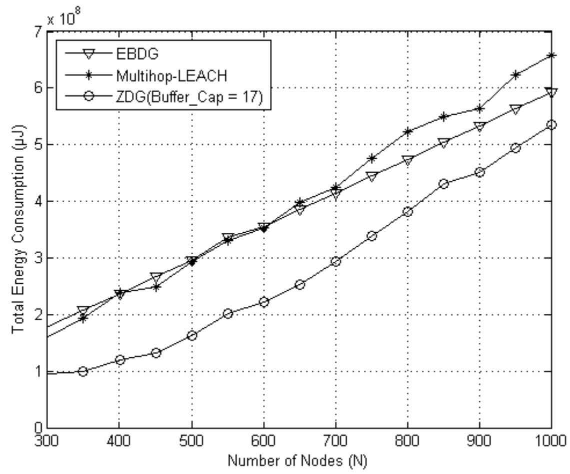

Fig. 5.2: Comparison of the energy consumption with EBDG and Multihop-LEACH for $Buffer\_Cap = 17$.

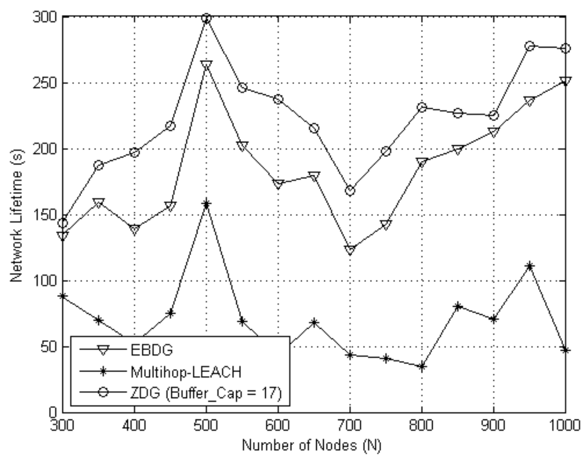

Fig. 5.3: Comparison of the network lifetime with EBDG and Multihop-LEACH for $Buffer\_Cap = 17$.

32.01% and 34.01% less energy (in average) than that of EBDG and Multihop-LEACH respectively.

The Fig. 5.3 depicts the network lifetime. As shown in Fig. 5.3, the network lifetime appears to be "fluctuating" across the cases. As nodes are randomly deployed in each simulation run, depending on the relative distance between the sensor node and the sink, one of the nodes may run out of energy early. Thus, the network lifetime varies in each configuration. Still, as shown in the figure, our ZDG network lasts 22.69% longer (in average) than EBDG. The network lifetime of our method elongates 2.6 times (in average) than that of Multihop-LEACH.

The Fig. 5.4 illustrates the throughput. As shown in Fig. 5.4, the through-

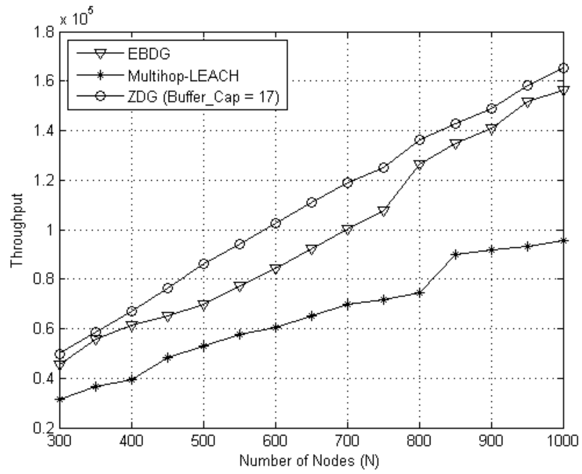

Fig. 5.4: Comparison of the throughput with EBDG and Multihop-LEACH for $Buffer\_Cap = 17$.

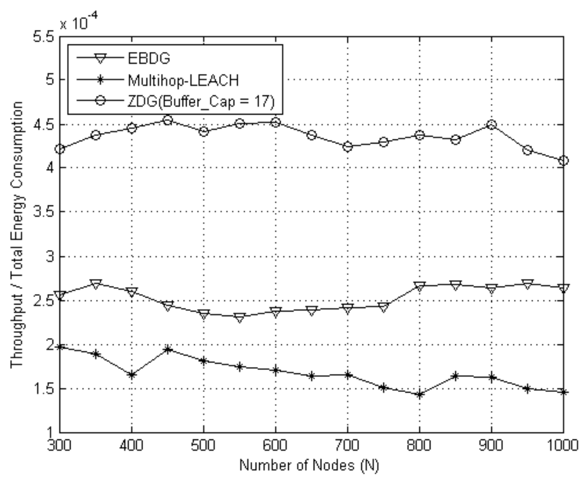

Fig. 5.5: Comparison of the throughput/energy consumption ratio with EBDG and Multihop-LEACH for $Buffer\_Cap = 17$.

put of ZDG is found superior than other methods because ZL and CHs can independently transmit their data by choosing the nearest zones and the system lifetime i.e., the time before all node die is much more prolonged than EBDG and multihop-LEACH. ZDG delivers 12.77% and 66.75% more packets (in average) than that of EBDG and Multihop-LEACH respectively to the sink.

The Fig. 5.5 is the summary of the ratio of throughput and total energy consumption. In Fig. 5.5, ZDG's throughput/energy consumption ratio is much better than the others. The ratio may vary because, nodes are deployed randomly and the number of hops needed to forward the data may vary and it may cause variation in energy consumption and the throughput. The result indicates that

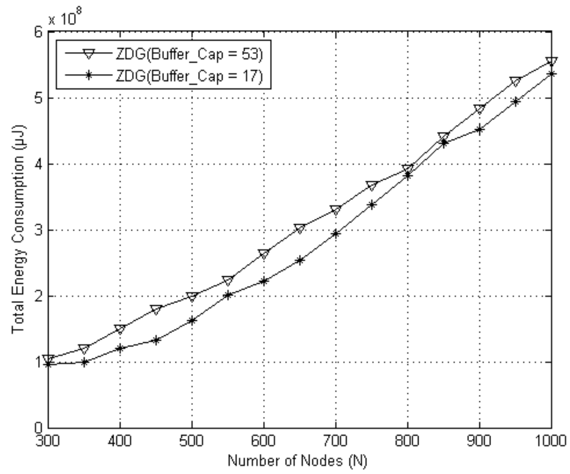

Fig. 5.6: Comparison of the energy consumption of ZDG for $Buffer\_Cap = 17$ and $Buffer\_Cap = 53$.

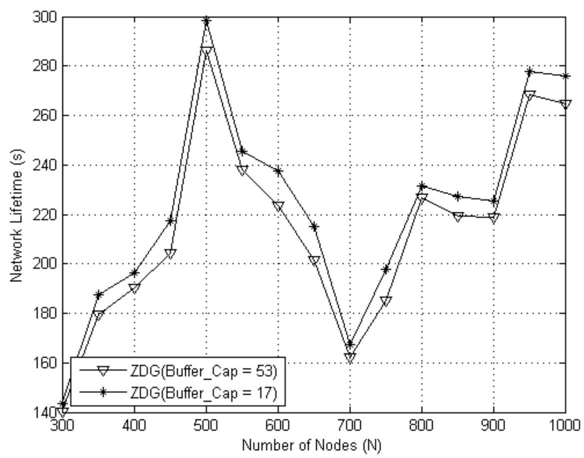

Fig. 5.7: Comparison of the network lifetime of ZDG for $Buffer\_Cap = 17$ and $Buffer\_Cap = 53$.

ZDG's throughput/total energy consumption ratios are 42.83% and 62.10% higher (in average) than that of EBDG and Multihop-LEACH respectively.

Fig.s 5.6, 5.7 and 5.8 show the comparison of the above metrics when the value of $Buffer\_Cap$ is 53. As portrayed in Fig.s 5.6, 5.7 and 5.8, ZDG for $Buffer\_Cap = 17$ shows better performance than for $Buffer\_Cap = 53$ in all cases. The reason is that when $Buffer\_Cap$ is 53, more data forwarding is needed which causes more energy consumption in sensor nodes (compared to when $Buffer\_Cap$ is 17) that degrades network lifetime and packet delivery efficiency. ZDG for $Buffer\_Cap = 53$ consumes 11.57% more energy (in average) than ZDG for $Buffer\_Cap = 17$. The network lifetime and throughput of ZDG

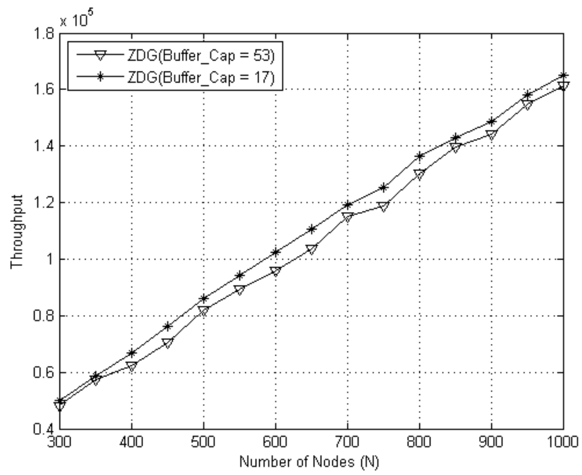

Fig. 5.8: Comparison of the throughput of ZDG for $Buffer\_Cap = 17$ and $Buffer\_Cap = 53$.

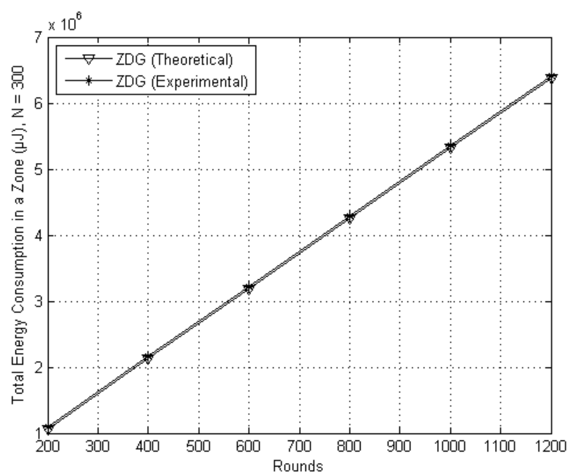

Fig. 5.9: Energy consumption per zone of ZDG.

for $Buffer\_Cap = 53$ are degraded by 4.21% and 4.54% respectively (in average) than that of ZDG for $Buffer\_Cap = 17$. Still, our method significantly outperforms the other methods.

To validate our theoretical analysis Equations 4.16 and 4.17, we compare the experimental results with the computed results. We see in Fig. 5.9 the total energy consumption for 300 sensor nodes with the similar parameters from Table 5.1. Here the results are from 200 rounds to 1200 rounds runs. We find that the experimental results is the similar to the theoretical results of ZDG. The characteristics and obtained results of both curves are almost the same which validate the derived equations.

The detail experimental results and variance calculations of EBDG, Multihop-LEACH and ZDG are listed in appendix A.

Chapter 6

Conclusion and Future Work

In this chapter, we give the summary of our research and a direction of the future extension.

6.1 Conclusion

In this thesis, we study different existing strategies to reduce energy consumption WSNs while data gathering. We identify some major limitations of those techniques. This thesis addresses the problem of energy limitations affecting the network lifetime. We develop a load balanced strategy for data collection and use the shortest path approach to increase the throughput of a static WSN.

To handle the challenges of data gathering, we present an energy efficient data gathering approach ZDG which reduces energy consumption, increases network lifetime and packet delivery. As a part of our strategy we partition the network into some zones based on geographic location. The tasks of data collection and aggregation are distributed among the nodes of the zones which resist the nodes from going to the bottleneck state by minimizing the data aggregation latency. To handle the buffer limitation problem of the sensor nodes, we divide again each zone into clusters. We select multiple cluster heads in the zone to handle the buffer limitation and to reduce load. In this thesis, we provide the theoretical analysis for our model. We simulate our technique with OMNeT++ [19]. Simulation results depict that about thirty percent energy consumption is saved in our method compared to the other popular zone based methods. We find that our method improves network lifetime and provide the higher throughput or packet delivery compared to those popular methods. Our method has better packet delivery compared to these method also. Finally, we validate our theoretical result by comparing the experimental results. We find that both results are the same.

6.2 Future Work

We would like to do some experiments in a heterogeneous sensor network where sensors have different energy profiles and different attributes. With some minor modifications, our method can be applicable for the heterogeneous network also. We would like to apply our method for mobile sensor network or a network with a mobile sink.

Bibliography

[1] Al-Refai H., Al-Awneh A., Batiha K., Ali A. A., and Rahman Y. M. E., "Efficient routing leach (er-leach) enhanced on leach protocol in wireless sensor networks," *International Journal of Academic Research*, vol. 3, no. 2, pp. 42–48, 2011.

[2] Arp J., "End-to-end acknowledgement for data collection in wireless sensor networks," Master's thesis, University of New Brunswick, 2010.

[3] Chippa M. K., "Performance of tree-based data collection in wireless sensor systems," Master's thesis, University of Akron, 2011.

[4] Cormen T. H., Leiserson C. E., Rivest R. L., and Stein C., *Introduction to Algorithms*. The MIT Press, 2001.

[5] Cricket Project Group, "The cricket indoor location system," http://nms.lcs.mit.edu/projects/cricket. Last visited May 17, 2011.

[6] Crossbow, Mica2 Wireless Measurement System Datasheets, http://www.xbow.com/Product/Products_pdf_files/Wireless_pdf. Last visited October 12, 2011.

[7] Euclidean Distance Measurement, http://en.wikipedia.org/Wiki/ Euclidean_distance. Last visited June 25, 2011.

[8] Heinzelman W. R., Chandrakasan A., and Balakrishnan H., "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[9] Heinzelman W. R., Chandrakasan A., and Balakrishnan H., "Energy-efficient communication protocol for wireless micro-sensor networks," in *Proc. of the*

Hawaii International Conference on System Science. Hawaii, USA, 2000, pp. 1–10.

[10] Heinzelman W. R., "Application-specific protocol architectures for wireless networks," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[11] Hill J., "System architecture for wireless sensor networks," Ph.D. dissertation, University of California, 2003.

[12] Khan Z. A., Belleudy C., and Auguin M., "Analytical model for energy consumption analysis in grid based wireless sensor networks," in *Proc. of IEEE International conference on New Technologies, Mobility and Security*. Cairo, 2009, pp. 1–5.

[13] Kim K., Lyo C. H., Moon S. S., and Youn H., "Tree-based clustering (TBC) for energy efficient wireless sensor networks," in *Proc. of IEEE International Conference on Advanced Information Networking and Applications workshops*. Perth, WA, 2010, pp. 680–685.

[14] Li X., Lu K., Santora N., Simplot-Ryl I., and Stojmenovic I., "Alternative data gathering schemes for wireless sensor networks," in *Proc. of International Conference on Relations, Orders and Graphs: Interaction with Computer Science*. Mahdia, Tunisia, 2008, pp. 577–586.

[15] Liang C., Lin J., and Li C., "Steiner points routing protocol for wireless sensor networks," in *Proc. of IEEE International Conference on Future Information Technology*. Busan, 2010, pp. 1–5.

[16] Lotfinezhad M., Liang B., and Sousa E., "Adaptive cluster-based data collection in sensor networks with direct sink access," *IEEE Transactions on Mobile Computing*, vol. 7, no. 7, pp. 884–897, 2008.

[17] Lu H., Lie J., and Wang G., "A novel energy efficient routing algorithm for hierarchically clustered wireless sensor networks," in *Proc. of IEEE International Conference on Frontier of Computer Science and Technology*. Shangai, 2009, pp. 565–570.

[18] Nguyen L. T., Defago X., Beuran R., and Shinoda Y., "An energy efficient routing scheme for mobile wireless sensor networks," in *Proc. of IEEE International Symposium on Wireless Communication Systems.* Reykjavik, 2008, pp. 568–572.

[19] OMNET++ User Manual Version 3.0, http://www.omnetpp.org/doc/manual /usman.html. Last visited August 06, 2011.

[20] Quan S. G. and Kim Y. Y., "Fast data aggregation algorithm for minimum delay in clustered ubiquitous sensor networks," in *Proc. of IEEE International Conference on Convergence and Hybrid Information Technology.* Daejeon, 2008, pp. 327–333.

[21] Renner B., "Energy-efficient tdma schedules for data-gathering in wireless sensor networks," Master's thesis, Hamburg University of Technology, 2008.

[22] Rickenbach P. V., "Energy-efficient data gathering in sensor networks," Ph.D. dissertation, ETS Zurich, 2008.

[23] Sensor Network Research Group at Louisiana State University, "Simulating wireless sensor networks with omnet++," http://bit.csc.lsu.edu/sensor_web/final_papers/SensorSimulator-IEEE-Computers.pdf. Last visited September 19, 2011.

[24] Shang F. and Lei Y., "An energy-balanced clustering routing algorithm for wireless sensor network," *Wireless Sensor Network*, vol. 2, pp. 777–783, 2010.

[25] VigilNet Project Group, "Vigilnet: An integrated sensor network system for energy-efficient surveillance," http://www.cs.virginia.edu/control/SOWN/ index.html. Last visited May 29, 2011.

[26] Vukobratovic D., Stefanovic C., Crnojevic V., Chiti F., and Fantacci R., "Rateless packet approach for data gathering in wireless sensor networks," *IEEE Journals in Selected Area in Communications*, vol. 28, no. 7, pp. 1169–1179, 2010.

[27] Wang C. and Wang C., "A concentric data aggregation model in wireless sensor network," in *Proc. of Progress In Electromagnetics Research Symposium.* Beijing, China, 2009, pp. 436–441.

[28] Wang F. and Liu J., "Networked wireless sensor data collection: Issues, challenges, and approaches," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 4, pp. 673–687, 2010.

[29] Weihua G., Liu Z., and Wu G., "An energy-balanced transmission scheme for sensor networks," http://www.cens.ucla.edu/sensys03/proceedings/p300-guo.pdf. Last visited July 11, 2011.

[30] Weng H., Chan H., Wu E. H., and Chen G., "A 2-approximation double tree algorithm for correlated data gathering in wireless sensor networks," in *Proc. of IEEE Global Telecommunications Conference*. Washington, DC, 2007, pp. 898–902.

[31] Wu Y., Fahmy S., and Shroff N. B., "On the construction of a maximum-lifetime data gathering tree in sensor networks: Np-completeness and approximation algorithm," in *Proc. of IEEE Conference on Computer Communications*. Phoenix, AZ, 2008, pp. 356–360.

[32] Xiangning F. and Yulin S., "Improvement on leach protocol of wireless sensor networks," in *Proc. of IEEE International Conference on Sensor Technologies and Applications*. Valencia, 2007, pp. 260–264.

[33] Yasssein M. B., Al-zou'bi A., Khamayseh Y., and Mardini W., "Improvement on leach protocol of wireless sensor network (VLEACH)," *International Journal of Digital Content Technology and Its Applications*, vol. 3, no. 2, pp. 132–136, 2009.

[34] Ye F., Luo H., Cheng J., Lu S., and Zhang L., "A two-tier data dissemination model for large-scale wireless sensor networks," in *Proc. of ACM Intenational Conference on Mobile Computing and Networking*. Atlanta, 2002, pp. 148–159.

[35] Zhang H. and Shen H., "Balancing energy consumption to maximize network lifetime in data-gathering sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 10, pp. 1526–1539, 2009.

[36] Zu-jue C., Shao-qing L., and Xi-liang Z., "Dynamic grid switch in multi-sink mobile sensor networks," in *Proc. of IEEE International Conference on Net-*

works Security Wireless Communications and Trusted Computing. Wuhan, Hubei, 2010, pp. 66–70.

Appendix A

Simulation Runs

Table A.1: **Experimental Results of EBDG (Average)**

Number of Nodes	EBDG		
	Total Energy Consumption (μJ)	Network Life-time (s)	Throughput
300	177606287	134.2	45397
350	207307156	159.2	55761
400	237009562	138.8	61459
450	266704260	156.9	64982
500	296407836	263.3	69796
550	336105981	202.4	77503
600	355803190	173.3	84419
650	385505672	179.6	92229
700	415205863	123.4	100042
750	444906000	142.3	107857
800	474609404	189.5	126289
850	504306849	199.6	134834
900	534008635	212.8	141034
950	563705689	236.2	151663
1000	593403792	251.3	156300

Table A.2: **Experimental Results of Multihop-LEACH (Average)**

Number of Nodes	Multihop-LEACH		
	Total Energy Consumption (μJ)	Network Lifetime (s)	Throughput
300	159510885	88.0	31452
350	194038108	69.8	36753
400	238327082	51.8	39476
450	248158312	75.4	48245
500	293327412	158.6	53119
550	331355740	69.0	57887
600	353095214	43.2	60354
650	398244324	67.8	65127
700	424444521	43.2	69897
750	476161344	41.0	71678
800	522230464	34.8	74598
850	549597798	80.6	89953
900	564483541	70.8	91659
950	623647594	111.0	93278
1000	658135845	47.0	95403

Table A.3: **Experimental Results of ZDG for *Buffer_Cap* = 17 (Average)**

Number of Nodes	ZDG ($Buffer\_Cap = 17$)		
	Total Energy Consumption (μJ)	Network Lifetime (s)	Throughput
300	95742197	143.6	49934
350	99554098	187.3	58526
400	120325372	196.6	66786
450	132479068	217.5	76192
500	161792364	298.5	86009
550	200713354	245.7	94267
600	221893529	237.5	102514
650	253132562	215.0	110772
700	294639690	167.6	119034
750	338186570	197.8	125175
800	381509834	231.5	136438
850	429834465	227.1	142789
900	452076026	225.2	148680
950	494712220	277.8	158104
1000	535262327	275.8	165201

Table A.4: **Results of Energy Consumption per Zone of ZDG (Average)**

Rounds	Energy Consumption per Zone (μJ)	
	ZDG (Theoretical)	ZDG (Experimental)
200	1063802	1087889
400	2127604	2173268
600	3191406	3222992
800	4255208	4287974
1000	5319010	5363786
1200	6382812	6413180

Table A.5: **Experimental Results of ZDG for *Buffer_Cap* = 53 (Average)**

Number of Nodes	ZDG ($Buffer\_Cap = 53$)		
	Total Energy Consumption (μJ)	Network Lifetime (s)	Throughput
300	103792114	140.4	48110
350	120454692	179.7	57354
400	150845369	190.1	62298
450	179871763	204.3	70544
500	199481367	286.2	82116
550	223413250	237.9	89373
600	264813378	223.5	95915
650	303132566	201.6	103782
700	329989652	162.0	115074
750	368346565	185.2	118979
800	391509834	226.8	130331
850	442035867	219.4	139714
900	483271625	218.9	144187
950	525912841	268.5	154625
1000	554925365	264.8	161407

Table A.6: **Variance Calculation of EBDG**

Number of Nodes	EBDG		
	Total Energy Consumption (μJ^2)	Network Life-time (s^2)	Throughput
300	1183	0.21	32.25
350	1126	0.15	35.25
400	1175	0.16	30.75
450	1175	0.13	34.00
500	1169	0.24	34.00
550	1157	0.16	26.48
600	1173	0.16	34.50
650	1182	0.19	30.50
700	1167	0.11	18.75
750	1183	0.18	33.19
800	1176	0.20	30.75
850	1157	0.24	30.69
900	1141	0.17	33.00
950	1136	0.28	13.44
1000	1157	0.27	31.61

Table A.7: **Variance Calculation of Multihop-LEACH**

Number of Nodes	Multihop-LEACH		
	Total Energy Consumption (μJ^2)	Network Life-time (s^2)	Throughput
300	1095	0.22	14.00
350	1023	0.38	15.23
400	912	0.18	20.36
450	1108	0.21	14.86
500	1110	0.22	12.11
550	1020	0.21	26.13
600	1100	0.20	31.23
650	1075	0.16	17.57
700	970	0.20	18.28
750	1093	0.22	31.09
800	1137	0.20	22.25
850	1012	0.17	21.69
900	988	0.18	19.83
950	1134	0.12	24.64
1000	1147	0.12	32.42

Table A.8: **Variance Calculation of ZDG for *Buffer_Cap*= 17**

Number of Nodes	ZDG ($Buffer\_Cap = 17$)		
	Total Energy Consumption (μJ^2)	Network Lifetime (s^2)	Throughput
300	1105	0.16	19.50
350	1130	0.10	18.36
400	1075	0.16	21.51
450	1082	0.18	23.50
500	1115	0.31	24.44
550	1104	0.20	16.23
600	1089	0.13	27.94
650	1153	0.19	19.44
700	1167	0.10	19.50
750	1174	0.39	27.32
800	1091	0.21	18.36
850	1127	0.22	25.12
900	1168	0.18	30.26
950	1123	0.30	21.41
1000	1117	0.23	16.73

Table A.9: **Variance Calculation of ZDG for *Buffer_Cap* = 53**

Number of Nodes	ZDG ($Buffer\_Cap = 53$)		
	Total Energy Consumption (μJ^2)	Network Lifetime (s^2)	Throughput
300	1133	0.11	24.34
350	1126	0.23	27.41
400	1161	0.26	19.78
450	1090	0.26	27.29
500	1115	0.35	31.89
550	1128	0.30	16.41
600	1119	0.23	22.58
650	1145	0.22	31.74
700	1088	0.31	33.12
750	1079	0.33	27.46
800	1124	0.17	28.57
850	1167	0.19	19.12
900	1138	0.28	30.26
950	1027	0.25	30.41
1000	1016	0.23	16.73