ANALYSIS, DESIGN AND OPTIMIZATION OF ENERGY EFFICIENT PROTOCOLS FOR WIRELESS SENSOR NETWORKS

HOANG DUC CHINH

NATIONAL UNIVERSITY OF SINGAPORE

2013

ANALYSIS, DESIGN AND OPTIMIZATION OF ENERGY EFFICIENT PROTOCOLS FOR WIRELESS SENSOR NETWORKS

HOANG DUC CHINH
(B.Eng., Hanoi University of Technology, Vietnam)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
2013

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Hoang Duc Chinh
29 August 2013

Acknowledgments

I wishes to record my deep sense of gratitude to my supervisor, Assoc. Prof. Sanjib Kumar Panda, who has introduced the present area of work and guided in this work. My thesis supervisor, Assoc. Prof. Sanjib Kumar Panda has been a source of incessant encouragement and patient guidance throughout the thesis work. I am extremely grateful and obliged to Dr. Rajesh Kumar for his intellectual innovative and highly investigative guidance to me for my project. I wish to express my warm and sincere thanks to the laboratory officers, Mr. Y. C. Woo, and Mr. M. Chandra of Electrical Machines and Drives Lab. NUS, for their readiness to help on any matter. The author's warmest thanks go to the fellow research scholars in Electrical Machines and Drives Lab for all the help to make my stay more enjoyable and beneficial. The author wishes to convey special thanks to Dr. Yen Kheng Tan, Mr. Parikshit Yadav for their valuable discussions on the design and development of my project and their constant help and suggestions in many aspects of my research works. My heartfelt gratitude goes to Dr. Haihua Zhou, Dr. Satyanarayan Bhuyan, Dr. S.K. Sahoo, Mr. Bhuneshwar Prasad and Krishnanand Kaippilly Radhakrishnan for their supportive and inspiring comments during my study.

I owes so much appreciation for many warm-hearted, and wonderful friends inside and outside of the NUS campus. Thanks to my old classmates, Nam Hoai Pham, Dr. Xuan Loc Nguyen, Dr. Thanh Long Vu, Tran Duong, Van Nghiem, Hoa Nguyen and Tuan Dung Phan for their encouragement and help for my PhD application. I am truely grateful to Muraliraj S/O Rajoo Devaraj and Ching Kuan Thye for their help to develop the software design of my project. I would like to say thank to Hien Nguyen, Hien La for their help to proofread my thesis. Also, I will cherish the friendship with Son Le, Luong Ha, Nghia Cao and Huong Nguyen and all the friends who take care of me and support me.

I have been deeply touched by endless love and boundless support by my family. I would like to thank to my sister, Kieu Ngan Hoang, my cousin, Thanh Dinh Khac, my grand mother, Le Bui, my grand father, Phach Dinh Khac, and my fiancee, Chuc Nguyen Thanh for all their love and support. I am indebted to my parents, Lan Dinh Thuy and Binh Hoang Duc, for everything that they have given to me. They have stood by me in everything I have done, providing constant support, encouragement, and love. I wish to dedicate this thesis for their love and support.

Contents

Summary

The recent advances in information and communication technologies enable fast development and practical applications of wireless sensor networks (WSNs). The operation of the WSNs including sensing and communication tasks needs to be planned properly in order to achieve the application-specific objectives. The WSNs consist of a number of sensor nodes equipped with microprocessor, wireless transceiver, sensing components and energy source. These sensor nodes operate as autonomous devices to perform different tasks including sensing, communication and data processing. As the members of a network, the sensor nodes are required to cooperate with each other to perceive environmental parameters and transfer data.

Commonly, the sensor nodes are left unattended in the environment after being deployed with limited resources such as computational ability, memory and energy. In order to serve for a long lifespan, the resources, especially energy, need to be utilized appropriately. Efficient energy usage is an essential requirement for each individual node as well as for the overall network.

A number of energy efficient protocols have been proposed in the literature. The cluster-based protocol is one classification which has the advantage of scalability, efficient communication and energy savings. This protocol organizes the

network into clusters, each cluster has one cluster head (CH) that gathers and aggregates data from all the cluster members, and then send to a base station (BS). Hence, the amount of transferred data is reduced that conserves the energy. The tree-based protocol is another type of protocols that supports multihop communication. Energy efficiency may not be as high as the cluster-based protocol, but it may be the more suitable option when the sensor nodes' communication range is not large enough to reach the destination, i.e. the BS, in one hop. In both of these protocols, data aggregation is an essential technique that enables significant decrease of data packets being sent and save large amount of energy consumption for data transfer. Optimization methods have been applied to improve the performance of these protocols during the organization of the network, i.e. cluster formation and cluster head selection in cluster-based protocols or tree construction and aggregation node selection in tree-based protocols. However, in some of the methods, only local optimum can be achieved. Furthermore, most of the protocols using optimization algorithms are investigated in simulation and not yet developed for real-life applications.

This thesis analyses the power consumption of wireless sensor nodes in practice for performing different tasks. The energy sources for sustaining the sensor nodes operation are also investigated. These sources include renewable energy sources like a thermal energy harvesting source and finite energy source like battery. The energy aware CH selection scheme for a small scale WSN consisting of one cluster is developed and evaluated in both simulation and real-time operation.

For a larger scale WSNs, three energy efficient protocols are proposed in this thesis. First, a Fuzzy C-Means cluster-based protocol (FCMCP) is designed in order to achieve a more uniform formation of clusters when compared to a typical cluster-based protocol such as LEACH. The FCMCP adopts Fuzzy C-Means (FCM) algorithm to organize sensor nodes into cluster with the expectation that the mean distances between these nodes and the centroid of each cluster is minimized. Therefore, the number of send nodes in each cluster is more equal, and traffic load is balanced at the CH, and the communication distance of the cluster members is reduced, which means less energy consumption for data transfer is required. Second, further improvements in the Fuzzy C-Means cluster-based protocol are carried out with the consideration of energy awareness into the cluster formation and cluster head selection. An objective function that attempts to optimize the mean distance of each cluster from its sensor nodes and simultaneously selects the best CH in terms of energy efficiency is formulated. In order to solve this problem efficiently, a novel evolutionary algorithm, called Harmony Search Algorithm (HSA) is adopted. The HSA mimics the process of improvising music. It is able to obtain the best fitness value with fast convergence in this problem when compared to other evolutionary algorithms such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). A general framework is also proposed for designing and implementation of the centralized cluster-based protocols for real-time WSNs with the support of various optimization methods. It is later applied to develop the cluster based protocols using FCM and HSA, called FCMCP and HSACP, on a hardware test-bed. The experiments are conducted to validate the energy efficiency and the ability of extending network lifetime of these two protocols.

Third, a tree-based protocol is proposed with the adoption of the nature-inspired optimization method, called Intelligent Water Drops (IWD) algorithm. The IWD algorithm imitates the flow of water drops in the streams or rivers that attempt to find the shortest way with fewer obstacles to reach the destinations like lakes or oceans. It is similar in the case of WSNs when data packets generated from some data source nodes look for the best routes that lead to the sink or base station. When being applied in the WSN, IWD algorithm optimally establishes a data aggregation tree that consists of minimum number of nodes with the aggregation nodes chosen as near to the data sources as possible. An improvement of the basic IWD algorithm is also carried out that can enhance the probability of finding the best aggregation nodes. The evaluation of the tree-based protocol using IWD is performed in simulation and compared with a well-known nature-inspired optimization method called Ant Colony Optimization (ACO). It is shown that with the application of IWD, the energy consumption for transferring data can be reduced and the network lifetime can be extended.

The computational as well as experimental results presented in the thesis have proved the better performance of the WSNs, i.e. longer network lifetime, when the proposed energy conservation mechanisms and energy efficient protocols are employed. It is also shown that the proposed centralized cluster-based protocols like FCMCP and HSACP are able to be practically implemented and operate the real-time WSNs efficiently.

At this stage, the study of network operation mainly focus on lifetime extension. However, the quality of service (QoS) is also required. Consideration of QoS introduces more dimensions to the optimization problems of the WSN that needs to be further investigated. Supporting services like localization have not been studied. The implementation of localization service enables fully autonomous operation of the sensor node. However, it also requires more resources, thus efficient methods need to be developed. The renewable energy sources is investigated in the thesis for individual nodes. Incorporation of these sources in the operation of the proposed protocols have not been done and is left for the future work. In additions, the investigation of a protocol using IWD algorithm for real-time operation can be also carried out in the future work.

List of Tables

List of Figures

List of Symbols

E_{Tx}	Energy consumption of a sensor node for transmitting data
E_{Rx}	Energy consumption of a sensor node for receiving data
E_{elec}	Energy consumption for operating transceiver circuit
E_{fs}	Free space model of the energy consumption for transmitting one bit data
E_{fs}	Multipath fading model of the energy consumption for transmitting one bit data
P_p	Power consumption for data processing of the microcontroller
P_c	The average power consumption of the radio component
Q	The heat flow between the body with a core temperature and the ambient air
P_{tx}	Transmission power of the sensor node
V_{th}	Threshold voltage representing the residual energy of the sensor node
E_{da}	Energy consumption of a cluster head for data aggregation

$T_{ReCluster}$	Time period to recluster the network
T_{DataTx}	Time period to transmit data before dropping
$T_{Sensing}$	Sample time for performing sensing task
T_{Cycle}	Time interval between two data transmission
T_{DataRx}	Maximum time period to receive data of a cluster head
$T_{DataAgg}$	Maximum time to aggregate data a cluster head
$T_{RadioOn-CH}$	Maximum time to keep radio on for sending one data packet at a cluster head
$T_{RadioOn-CM}$	Maximum time to keep radio on for sending one data packet at a cluster member
$soilLoad^{IWD}$	The amount of soil an IWD carries or its soil load.
vel^{IWD}	The velocity at which it is moving
$soil(i,j)$	The soil on the bed of the edge between point i and point j
a_v, b_v, c_v	The parameters used to update the velocity
a_s, b_s, c_s	The parameters used to update the amount of soil
$time(i,j;vel^{IWD})$	The time for an IWD to move from point i to point j at the velocity vel^{IWD}
$HUD(i,j)$	The local heuristic function presenting the undesirability of an IWD to move from point i to point j
$p(i,j;IWD)$	The probability of an IWD to move from point i to point j
T^{IB}	The best solution found in each iteration

Acronyms

ACO	Ant Colony Optimization
ADV	ADVERTISEMENT
ASG	ASSIGNMENT
AGG	Aggregated data packet
ANNC	Announcement
APTEEN	Adaptive Threshold sensitive Energy Efficient sensor Network
B-MAC	Berkeley Media Access Control
BPK-means	Balanced Parallel K-means
BS	Base station
BTN	Baton
CNS	Center at the Nearest Source
CH	Cluster Head
CM	Cluster Member
CSMA	Carrier Sensing Multiple Access

CSMA/CA Carrier Sensing Multiple Access-with Collision Avoidance

DD Directed Diffusion

EAR Energy-aware Routing

FCM Fuzzy C-Means

FCMCP [Fuzzy C-Means Clustering Protocol]

FCA Fixed-CH Algorithm

GA Genetic Algorithm

GIT Greedy Incremental Tree

GPS Global Positioning System

GUI Graphic User Interface

HAL Hardware Adaptation Layer

HAN Home Area Network

HIL Hardware Interface Layer

HM Harmony Memory

HMCR Harmony Memory Considering Rate

HMS Harmony Memory Size

HPL Hardware Presentation Layer

HSA Harmony Search Algorithm

HSACP Harmony Search Algorithm based Clustering Protocol

IWD Intelligent Water Drop

LEACH Low Energy Adaptive Clustering Hierarchy

LEACH-C LEACH Centralized

MAC Medium Access Control

MTE Minimum Transmission Energy

NiMH Nickel-Metal Hydride

NWK Network

OS Operating system

OSI Open Systems Interconnection

PAMAS Power Aware Multi-access protocol with Signaling

PAR Pitch Adjusting Rate

PEGASIS Power-efficient Gathering in Sensor Information Systems

PSO Particle Swarm Optimization

PHY Physical

RSA Random Selection Algorithm

RSSI Receive Signal Strength Indicator

S-MAC Sensor-MAC

SMACS Self-Organizing Medium Access Control for Sensor Networks

SpeckMAC Speck Media Access Control

SPIN	Sensor Protocols for Information via Negotiation
SPT	Shortest Path Tree
SSA	Sequential Selection Algorithm
STEM	Sparse Topology and Energy Management
TDMA	Time Division Multiple Access
TEEN	Threshold sensitive Energy Efficient sensor Network
TEG	Thermoelectric Generator
TEH	Thermal Energy Harvesting
TRAMA	TRaffic-Adaptive Medium Access protocol
TMAC	Timeout-MAC
TSP	Traveling Salesman Problem
VbSA	Voltage-based Selection Algorithm
WBAN	Wireless Body Area Network
WPAN	Wireless Personal Area Networks
WSN	Wireless Sensor Network
Z-MAC	Zebra MAC

Chapter 1

Introduction

The demand of acquiring ambient knowledge to create smart environment results in the invention of new intelligent devices which have all the abilities of sensing, computation and communication. Through advanced networking protocols, these devices are connected to each other and form a so-called Wireless Sensor Network (WSN) [1].

The earliest generations of WSNs were used in military applications to carry out battle field surveillance or enemy tracking. However, modern technologies in wireless communication and digital electronics have enabled the development of small size, inexpensive and multifunctional sensor nodes, and thus make WSNs possible to be employed in other daily life application such as health care monitoring, home automation, disaster prediction, seismic structure monitoring, surveillance, etc [2–4]. Depending on the specific applications, WSNs can be deployed into any types of environmental field which are in many cases impossible for conventional wired sensor systems like forested areas, battle fields, coal mines, deep oceans, etc [5]. Each sensor node, as an autonomous device, perceives the physical parameters of the deployment field, processes the data, and then incorporates with data from other nodes in the network to transfer the crucial information to destinations. However, the sensor nodes are usually equipped with restricted resources. In order to successfully fulfill the aims of applications as long as possible, it is very challenging to operate the sensor nodes as well as the overall network efficiently.

1.1 Basis of Wireless Sensor Networks

A WSN can be simply defined as a network of devices denoted as nodes that can sense the environment and process the data gathered from the monitored field through wireless links; the information are later forwarded, directly or through multiple hops, to one or several sinks that can use it locally or to other networks through a gateway [1].

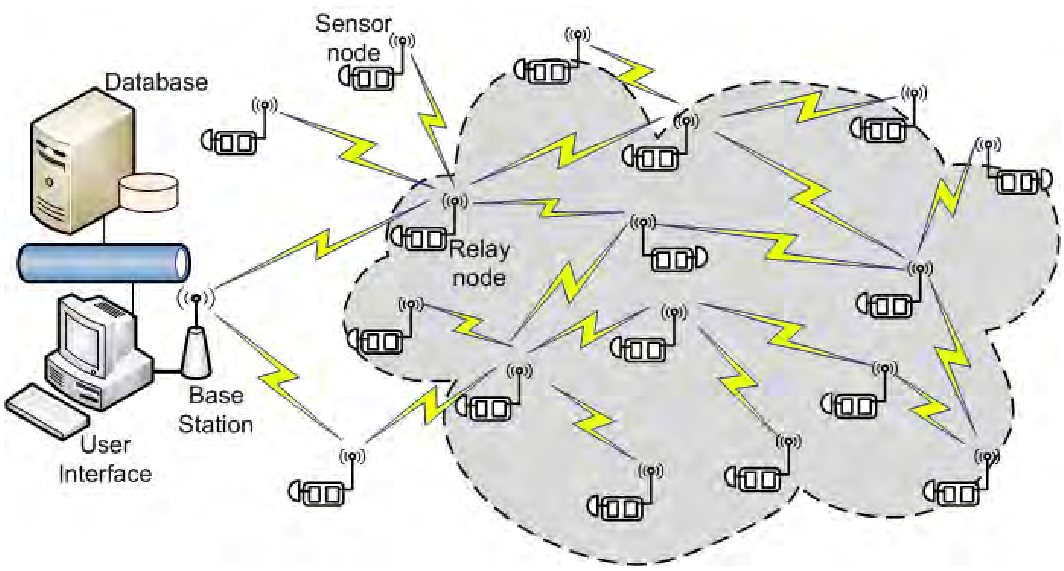


Figure 1.1: Sensor nodes deployed in the environment

A typical sensor network is illustrated in Fig. 1.1. As shown in Fig. 1.1, there are several types of nodes playing different roles in the network.

- **Base stations (BSs)** are the sinks or destinations that take the responsibility of collecting data from the WSNs. The received data can be either processed by the BS, then displayed on a user interface or stored in a database for later use.

- **Sensor nodes** are the data sources which generate data in the network. The sensor nodes are equipped with sensing devices to perform the measurement process. Measurement values perceived by the sensors are transmitted through the network to appropriate destinations.

- **Relay nodes** are the bridges between the sensor nodes and the BSs. These nodes, in many cases having the same architecture as the sensor nodes, are able to receive data packets from different sources and forward them to other nodes towards the destinations. In some scenarios, the relay nodes may perform processing task such as aggregating data before transferring them to the next nodes.

These nodes together would form a full functioning WSNs with the support of proper organisation mechanisms. The organisation mechanisms of the networks are identified as communication protocols which assist to maintain the connectivity of the sensor nodes and transferring data within the network in a reliable and efficient way.

1.1.1 Composition of single nodes

A sensor node is a fundamental unit in WSNs which is a complete embedded system with the functions of sensing, computation and communication. A typical sensor node consists of a sensor module, a microcontroller, a transceiver and a power supply as shown in Fig. 1.2 [6].

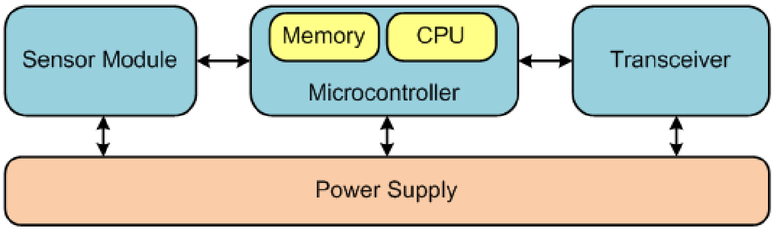


Figure 1.2: Typical architecture of a sensor node

a) Hardware components: The individual hardware components of a single sensor node are described below.

Sensor module - The actual interface to the physical world which can perceive physical parameters of the environment and convert them into digital values. Examples of sensor are temperature, light, humidity, vibration, acoustic or image sensors.

Microcontroller - A device to process the relevant data and manage the operation of other components. The microcontroller of the sensor node has usually limited capabilities of sensor nodes.

Transceiver - The wireless communication unit which is used to transmit and receive information from the network.

Power supply - Most of the sensor node's components require electrical energy for their operation; the energy is provided by a power supply. The power supply can be a finite energy source like battery and super capacitor, or a renewable energy source that is able to harvest energy from the ambient environmental energy sources such as solar, wind, vibration, etc.

In some application scenario, actuator can be added to control physical parameter in the deployment field.

b) Operating Systems

The execution of all the tasks of the sensor node such as sensing, data processing and communication, requires an embedded Operating system (OS) which has the responsibility of managing the concurrency of several processes, controlling and protecting the access to hardware resources, providing underlying OS services to applications and abstracting the hardware from the software applications [7]. Furthermore, for WSNs, the energy-efficient execution is a critical task, the OS needs to support for energy management that can turn on and off individual components or adjust the frequency of the system operation to achieve power saving meanwhile the objectives of the application are still qualified.

Due to the limitation in memory storage, computational ability and energy source, the OS for WSNs itself needs to be proficient, lightweight and easy to install and to reprogram. Besides, the application-specific nature of WSNs necessitates application-specific OSs. A programming model for sensor nodes based on an OS is illustrated in Fig. 1.3. The application layer is implemented on top of the model, uses services such as transmitting and receiving data packets, or acquiring the sensor measurements provided by the lower layer of the OSs.

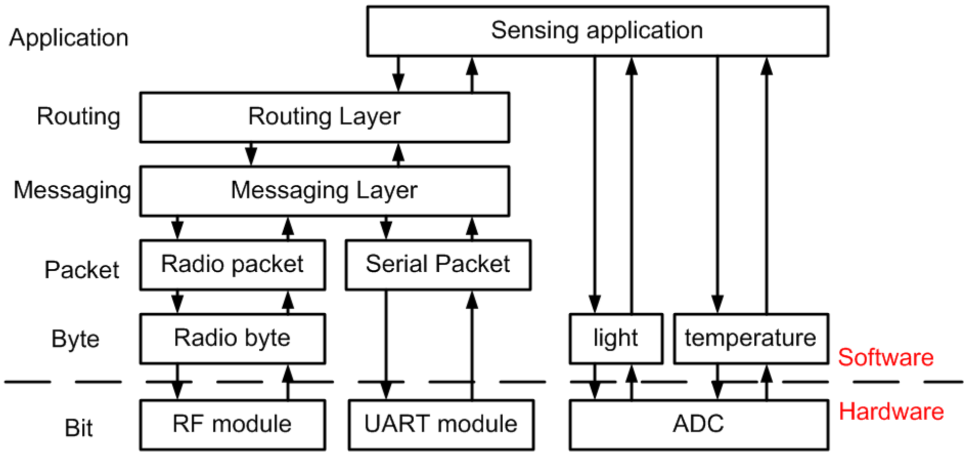


Figure 1.3: An example of programming model for Sensor Nodes (adapted from [8])

For the purpose of increasing portability and simplifying application development, the independence of the application layer from the hardware platform is required. The hardware abstraction in modern operating system is introduced for this purpose. Fig. 1.4 shows a three-tier hardware abstraction architecture in OS for WSNs. The top level of abstraction, Hardware Interface Layer (HIL), fosters portability by providing a platform-independent hardware interface; the middle layer, Hardware Adaptation Layer (HAL), promotes efficiency through rich hardware-specific interfaces, and the lowest layer, Hardware Presentation Layer (HPL), structures access to hardware registers and interrupts.

Several OSs following different approaches have been developed for WSNs. There exist three main architectural alternatives for implementing concurrent processing in terms of OS for WSNs which are process-based preemptive multithreading, event-based programming or both of these two [11]. Some of the OS for WSNs and their comparison are shown in Table 1.1

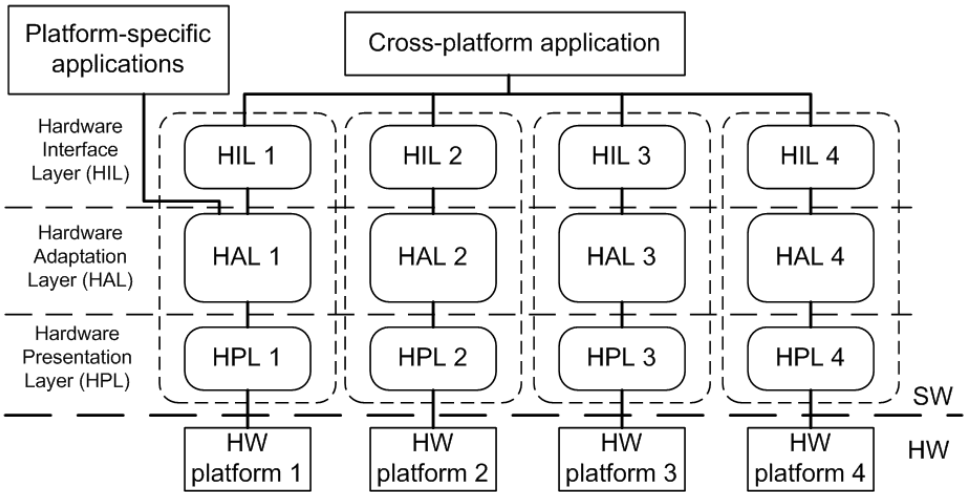


Figure 1.4: Hardware Abstraction Architecture in OS for WSN (adapted from [10])

Table 1.1: OSs for WSNs and their specifications

OS	Main approach	Code Memory (KB)	Data Memory (B)
TinyOS	event-based	3.4	226
SOS	event-based	20.0	1163
BTnodes OS	event-based	34.7	1029
Mantis	preemptive	~ 14	500
RETOS	preemptive	23.7	1125
Contiki	both	3.8	> 230

1.1.2 Communication protocols

Conventional networking protocols are abstractedly described in Open Systems Interconnection (OSI) model which includes seven layers [12]. In WSNs, protocol stack used by all the sinks and sensor nodes has the responsibility of combining power and routing awareness, integrating data with network protocols, controlling access to the wireless medium and promoting cooperative efforts of sensor nodes [6]. Protocol stack consists of five layers:

- *Application layer:* This layer supports the application software in application task management including sensing task and communication services, and provides the user with the functionalities to set up the network's configuration.

- *Transport layer:* The transport layer assists to maintain the flow of data for the network applications, and is especially needed when it is required to access to the system through internet or other networks.

- *Network layer:* The main objective of this layer is to route the data among sensor nodes. Network layer is necessary for multi hopping communication from the source to the sink.

- *Data-link layer:* The data link layer is responsible for the multiplexing of data streams, data frame detection, medium access and error control. The medium access control (MAC) protocol should have the following attributes: energy efficiency, scalable to node density, frame synchronization, fairness, bandwidth utilization, flow control and error control.

- *Physical layer:* The physical layer is responsible for frequency selection, carrier frequency generation, signal detection, modulation and data encryption.

Figure.1.5 illustrates the communication protocol stack of the Zigbee Standard [14] that is based on the IEEE 802.15.4 standard [15]. In this standard, the transport layer is implicit, protocol stack include four layers which are application layer, network layer, MAC layer and physical layer. The MAC layer and physical layer are defined by IEEE 802.15.4 standard for low power, low rate and short range communication network. Meanwhile Zigbee incorporates this standard to emerge a wireless mesh network standard that provides high reliability and more extensive range. In order to support the communication, several system services and techniques can be employed such as localization, synchronization, data aggregation and

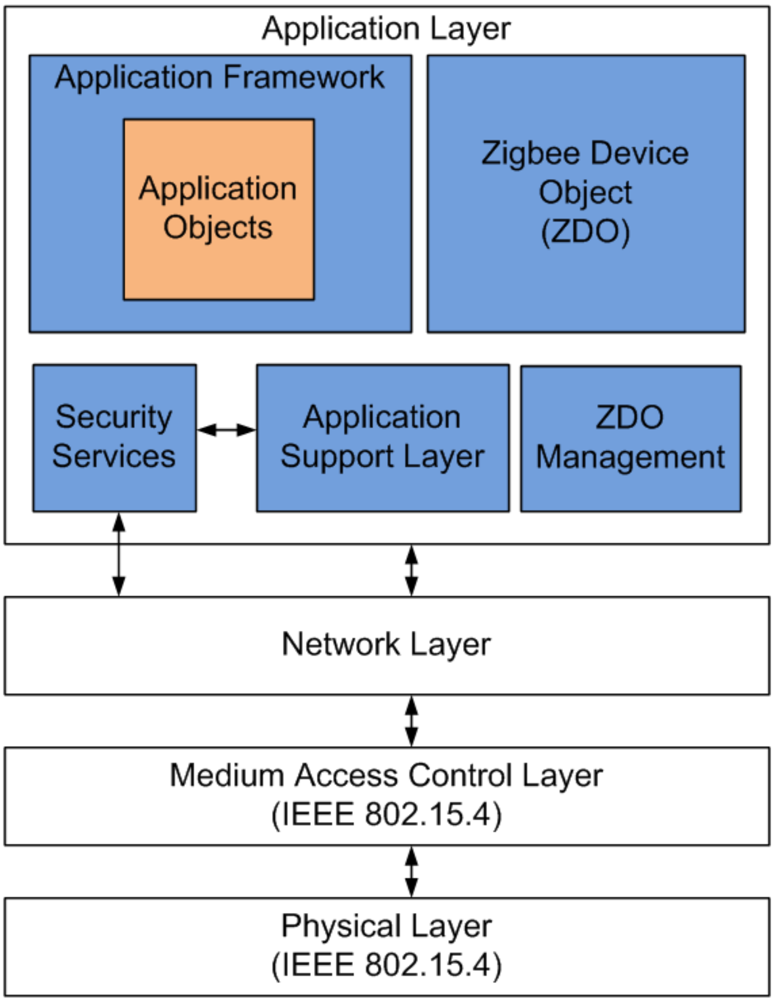


Figure 1.5: Network protocol stack in Zigbee Standard (adapted from [13])

compression, coverage and security. In WSNs, the performance of the communication protocol is tightly related to the energy efficiency. Thus high proficiency of energy usages is an essential requirement and also an objective of these services.

1.2 Motivation

The network lifetime is one of the most important critaria for evaluation the WSNs. The network can only fulfill the application-specific objectives as long as it is considered alive, but not after that [16]. After being deployed into the field, the sensor nodes are usually left to work independently with restricted resources, especially limited power supply such as battery or super capacitor. The sensor nodes need to utilize it resources wisely, and cooperate with other nodes in the network efficiently. This enables long operation of the WSNs, or in other words it can extend the network lifetime. Hence, optimizing the energy usage of each sensor node as well as the overall network is one of the most essential tasks when designing and planning WSNs operation.

Renewable energy sources such as solar or wind can be a solution to power the sensor nodes for a longer period of time. Although these sources can be considered as infinite sources, they are fluctuating and intermittent. For example, solar cell can harvest energy only during the day time or when it is exposed under sunlight but not in the shadow or in the night. Therefore, energy saving mechanism is always necessary to utilize the energy source efficiently so that the lifetime of each individual sensor node as well as the overall network can be prolonged.

The energy consumption of the sensor node is distributed to perform sensing, data processing and data transferring tasks. In general, data transferring within the network consumes most of the energy of sensor nodes [6]. Routing protocols play a crucial role in managing energy consumption of the WSNs, they define the route of transferring data packets from the sources to the destinations. In order to save energy, the routing protocols need to design in such a way that the route established by the protocols is able to transfer data efficiently. One technique that can be used to reduce energy consumption of data communication is data aggregation. Data aggregation technique enables the network to reduce the amount of data packets transferred by combining relevant data packets into a compressed one at a proper selected node before transmitting data. This technique can be applied in two categories of routing protocol: *cluster-based routing protocols* and *tree-based routing protocols*.

The cluster-based routing protocol is well recognized for its advantages of scalability and energy efficient communication. This protocol organizes sensor nodes into groups called clusters, each cluster has a Cluster Head (CH) that takes the responsibility of receiving and aggregating data from all the members and send the compressed packet to a BS [17]. However, to obtain proficient operation, the cluster-based routing protocol has to deal with the optimization in cluster formation, cluster head selection and data transmission in order to achieve efficient energy consumption, obtain high data throughput and prolong the network lifetime under resource constraints. Eventhough, many cluster-based protocols have been proposed in the literature such as [18, 47, 63–65], some protocols are not able to achieve optimal network organisation during the cluster formation and cluster head selection. For example, LEACH is one of the most well-known cluster-based protocols that has been shown to be energy efficient when compared with some conventional protocols [18]. However, the clusters formed by LEACH are not uniform, i.e. some clusters consist of a large number of sensor nodes whereas others have very less. The inequality of the number of sensor nodes within the clusters results in high traffic load and fast energy depletion at the CHs of the bigger clusters. Therefore, further improvement of these protocols is still needed with the support of novel optimization techniques in order to obtain global or near global optimum of network organisation.

In other scenarios, when the communication range of the sensor nodes is limited, it may require more than two hops, i. e. two intermediate nodes, to transfer data from the sources to the destination. Thus, a tree-based routing protocol with multihop communication and in-network data aggregation can be used in this case [19]. When the source nodes, that generate data in the network, find routes to the destination, i.e. a BS, a tree is established to connect all the source nodes to the BS via a set of intermediate nodes. Energy conservation can be achieved by applying data aggregation techniques at suitable nodes along the tree to reduce the amount of data packets transferred within the network. Hence, it decreases the communication burden of the network that is corresponding to less energy consumption. However, the construction of the tree as well as the selection of the aggregation nodes need to be optimized to achieve high energy efficiency when transferring data.

1.3 Problem Statement

Energy consumption is the main concern when designing and planning the operation of the WSNs to obtain the long lifetime. Thus, awareness of the residual energy, availability of harvesting renewable energy and configuration of power consumed by different components of the sensor nodes are very important. The investigations of the energy sources as well as the energy consumption are needed to serve the configuration of the network. It is also essential to study the efficient algorithms that assist the organisation and operation of the WSN efficiently in terms of energy. These aspects are formed into the following problems those are investigated in this thesis.

1.3.1 Energy conservation mechanisms for small scale WSNs

In a small scale WSNs such as Wireless Body Area Networks (WBANs) or Home Area Networks (HANs), there are a small number of sensor nodes deployed in a small area to perceive physical parameters and send data to a BS located far way from the deployment field. For example, in medical health care systems, a WBAN is used to monitor the condition of the patients. Sensor nodes of this WBAN, located in different places on the human body and grouped into a cluster, perceive information and transmit to a CH that later processes data and forwards to a BS, i.e. hospital servers or doctors' personal computer [9]. In most of the cases limited energy batteries are used to power the sensor nodes. The capacity of battery is proportional to its size and weight; tiny sensor nodes used in body area network to be carried by human require small form factor, hence solely dependent on battery may not be able to sustain the operation of the sensor node, especially the node that plays a role of a CH which consumes more energy to gather data and forward to a distant BS. An improvement of the sensor node's lifetime can be achieved by using thermal energy source which harvests energy from human body warmth. The principle is based on the difference between temperature of the body and that of the ambient environment. However, the amount of energy harvested is limited due to the small size of the harvester and the fluctuation of the energy source due to the change of the ambient temperature. An extra mechanism can be used to utilize the energy more efficiently among the sensor nodes of the networks that is to rotate the CH role among sensor node, the node with the highest energy level can be chosen as the CH. This mechanism guarantees that the CH has enough energy to perform the communication task including receiving data packets from all other nodes and transmit data to the BS. It also takes the benefit of the renewable energy source, since the nodes can balance between the energy consumed and energy harvested.

The mechanism of rotating CH can be also used in HANs. Particularly, a small WSN can be deployed in home environment for fire detection and danger alarm when there is presence of human being in the area of interest. The network includes one group of sensor nodes equipped with temperature and motion sensors measure the condition of the environment. Information form all the nodes are collected and compressed at the CH before forwarding to the BS. Power management within a cluster is required to prolong the network lifetime when the communication task is performed. A practical design and implementation of a dynamic selection scheme of a CH is needed to balance the energy usage among the sensor nodes. This scheme is based on the nodes' energy level by monitoring their battery voltage. Further improvement of energy savings is achieved by applying a time scheduling mechanism for data transmission of the nodes within the cluster. Here, only the CH needs to turn on its transceivers for a longer time to collect data sent from all the cluster members, meanwhile it takes a short time for member nodes to transmit information to their CH.

Another techniques that can be applied to reduce the energy consumption of the nodes is to adjust the transmission power based on the communication distance. The sensor nodes can reduce their transmission power properly to send data meanwhile the reliability of the transmission is guaranteed. A transmission scheme of a linear network is studied to evaluate the energy savings when the technique is applied.

1.3.2 Improvement of the network formation of cluster-based WSNs

The WSNs consists of a number of nodes, those are stationary after being deployed, are considered. The sensor nodes are employed to monitor some physical parameters of the area of interest. It might be highly co-related or duplicated data between the measurements collected by the sensor nodes. Hence, it is more efficient to reduce the amount of data packets transferred within the network that helps to save nodes' energy. In this case, cluster-based protocols can assist to operate the WSNs efficiently with the support of data aggregation techniques. Additionally, the network formation in cluster-based WSNs plays a crucial role to improve the energy savings of the networks. As mentioned earlier, a well-known cluster-based protocol like LEACH uses a random mechanism to select the CHs and then allocates the sensor nodes into clusters according to the distance from the nodes to the CHs. The random selection of CHs results in a non-uniform cluster formation. Some clusters may suffer high traffic load and the CHs need to consume a huge amount of energy. Therefore, it results in the fast energy depletion of these CHs. To improve the network formation, it is needed to apply efficient clustering algorithms.

One technique popularly used to cluster sensor network is k-means. In [21], the authors proposed a Balanced Parallel K-means (BPK-means) based clustering protocol based on k-means algorithm. The BPK-means protocol is developed to reduce energy consumption of the sensor node during communication with the CHs. However, with the random deployment of the sensor nodes in an unattended field, hard partitioning the network by means of using k-means algorithm may result in vagueness of classifying sensor nodes near the boundary of the clusters. Thus, the optimization of cluster formation is not obtained. Therefore, it is essential to develop a more innovative cluster-based protocol using a soft partitioning mechanism like FCM in order to further improve the creation of clusters in randomly deployed sensor networks WSNs.

1.3.3 Optimization of network formation and cluster head selection with nature-inspired optimization methods

In cluster-based WSNs, network organisation usually consists of two phases: *network formation* and *cluster head selection*. The FCM clustering algorithm performs network formation and selects the best CH among the nodes within each cluster. This method can achieve a good cluster formation, but may not obtain the optimal cluster head selection, since the optimization process is only based on the information of sensor nodes' position.

Additionally, the FCM algorithm, like K-Means clustering algorithm, starts with randomly initializing k cluster center points, and then nodes are assigned to the cluster which contains the nearest cluster center point. The solutions of these methods are highly sensitive to starting points and frequently converge to local optimum solution or diverge altogether. Furthermore, both the methods seek local optima. Thus starting the search in the vicinity of local optima will cause us to miss the global optima [22].

Recently, Harmony Search Algorithm (HSA) was developed by Geem et al. in 2001 [23] inspired by music improvisation process. The musician searches for a perfect state of harmony by adjusting the pitch. The effort to find the harmony in music is analogous to find the optimality in an optimization process. In other words, a musician's improvisation process can be compared to the search process in optimization. The pitch of each musical instrument determines the aesthetic quality, just as the objective function value is determined by the set of values assigned to each decision variable [23–26].

In this work, a cluster-based protocol which attempts to optimize the cluster formation and the cluster head selection simultaneously with the consideration energy efficiency. The application of such a nature-inspired optimization method like HSA is expected to find a global optimal or near optimal solutions for the organisation of the cluster-based WSNs.

The final aim is to build WSNs which can operate in optimal nodes with the support of the energy efficient routing protocols and are carefully validated and evaluated in a real time test-bed system.

1.3.4 Optimal construction of data aggregation tree

Another problem considered in this thesis is the construction of the optimal data aggregation tree for a densely deployed WSN to route data from different sources to a BS. This tree-based protocol is applied when the communication range of each sensor nodes is limited and it is required more than two hops to transfer data form the source to the destination. In order to achieve energy efficiency, the aggregation tree needs to be established in such a way that the junction nodes, where the routes from different sources meet, should be near to the sources and the total number of edges representing the direct communicating connection between the two nodes is minimized. A novel nature-inspired optimization algorithm, called IWD, is adopted with the expectation that the optimal or near optimal tree is achieved. This nature-inspired approach may provide a suitable mechanism for routing data in distributed systems like WSNs. The algorithm was first proposed by S. H. Hamed [27], which imitates the dynamic of river systems and the behaviours of water drops when they are moving in the river such as the variation of velocity, the change of soil in the bed of the river, the change of the flow direction, etc. The IWD algorithm changes the amount of soil on the paths, that water drops traversed. This variation depends on the velocity and the soil carried by the water drop, and it can be increased or decreased to attract or obstruct other water drops. This variation is different from the Ant Colony Optimization (ACO) algorithm, in which the pheromone is only increased to attract more or fewer ants following the discovered routes. The IWD algorithm has been employed to solve classical problems like Traveling Salesman Problem (TSP), N-Queens and Multiple Knapsacks problems [28]. It also can be used to solve optimization problems in other applications, e.g. robot path planning [29] and automitc multilevel thresholding for image segmentation [30]. The results show that IWD is capable to obtain near optimal solution for all these complicated problems in a finite number of iterations, and in some cases it performs better than ACO. Therefore, IWD has better balance between diversification and intensification, and it may have higher chance to find the optimal or near optimal route.

The basic IWD algorithm forms the aggregation tree with the re-enforcement of attracting other water drops in the next round of the tree construction at the junction nodes of the tree. However, it may happen that no junction node is found during the earlier rounds. Hence, the improvement of the basic algorithm is needed to enhance the construction process of the tree.

The basic IWD algorithm forms the aggregation tree with the re-enforcement of attracting other water drops in the next round of the tree construction at the junction nodes of the tree. However, this has limitation when no junction node is found during the earlier rounds. Thus, modification of the basic model IWD algorithm need to be carried out in this work in order to effectively apply the method for solving data aggregation optimization problem.

1.4 Contributions of the Thesis

- Investigation of energy conservation mechanisms for small scale WSNs has been carried out. Several energy conservation mechanisms have been studied including energy scavenging using thermal energy from human body, battery aware cluster head selection and transmission power adjustment. A thermal energy harvesting source is proposed and applied to power the sensor nodes for fall detection in health care medical systems. It successfully harvests energy from the warmness of human body based on the Seebeck's effect in order to power the sensor nodes. The electric voltage is generated when there is a difference between the human body temperature and the ambient temperature across two dissimilar materials. Furthermore, the battery-aware CH selection scheme is proposed and investigated with a small scale WSN.

- Energy efficient cluster-based protocols for WSNs is proposed. The thesis presents the design and analysis of centralized cluster-based routing protocols using different techniques such as FCM clustering and HSA to find the optimum of the WSNs structure. First, the FCM algorithm is applied to enhance the cluster formation when compared with a typical cluster-based protocol like LEACH. Then the nature-inspired optimization method, HSA is employed to solve a multiobjective problem of optimizing the cluster head selection and the cluster formation simultaneously. The performance of the protocol is studied and compared with the conventional methods and other evolutionary algorithms such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) [31]. These algorithms assist to form the clusters and choose CHs in such a way that the energy consumption of the network is reduced and thereby, network lifetime is extended.

- Design and implementation of cluster-based protocols using optimization methods on real-time systems have been presented. In order to investigate and apply the cluster-based protocols practically, it is essential to design and implement cluster-based protocols on a hardware test-bed. In this work, a framework for developing centralized cluster-based protocols has been proposed that can be applied to evaluate different clustering methods. The development of this type of protocols is based on the TinyOS [33], an embedded operating system developed for networking devices with limited energy and memory resources like wireless sensor nodes. The centralized category is selected to utilize the abundant resources and strong computation capability of the BS. Hence, it enables the validation of the optimization methods for network organisation like FCM and HSA. It is also first attempt to employ such a nature-inspired optimization method like HSA in the real-time wireless sensor systems. The experimental results show the successful application of FCM as well as HSA in the practical WSNs with high energy efficiency.

- A mechanism for constructing the optimal data aggregation trees for multihop WSNs using IWD algorithm has been developed. In the case of tree-based routing protocols, a problem of constructing optimal data aggregation tree for a multihop WSN is studied. The data aggregation tree needs to be established in such a way that it can aggregate as many data packets as possible at the earlier stage of each round of transmitting data in the WSN and transfer them via energy efficient routes. The IWD algorithm has been adopted to solve this problem with the expectation of finding the global optimum or near optimum. This is the very first attempt to apply this algorithm for constructing the routing in WSNs. A modification of the basic model IWD algorithm has been proposed in this work in order to effectively apply the method for solving data aggregation optimization problem. The computational experiments have been conducted to prove that the improved IWD and IWD algorithms can achieve good results comparable with ACO algorithm in terms of energy efficiency.

1.5 Thesis Organisation

This chapter introduced the fundamentals of WSNs, the motivation and problem statement of the research work, as well as the contributions.

Chapter 2 presents the related works in the literature which support the research work in this thesis. The basic of the energy saving techniques and the typical routing protocols are described. The optimization methods which have been used for supporting routing protocols and adopted in this thesis are also introduced.

Chapter 3 described the system architecture and energy model that are used to evaluate the performance of the proposed protocols. Different energy conservation mechanisms are proposed and analyzed in this chapter. The experiment of a WSN in medical health care systems is performed. The sensor nodes are equipped with a renewable energy source harvesting from human warmth, and an enhanced star topology WSN with the ability of rotating gateway among sensor nodes is applied. The initial development of cluster-based routing in real time hardware platform with associated techniques is also discussed. This includes the controlling of transmission power based on received signal strength indicator, making decision of cluster head selection and time scheduling of data communication within a cluster. These mechanisms are later used as components in the complete protocols proposed in the next part of the thesis.

Chapter 4 analyzes different routing strategies and introduces cluster-based protocols for WSNs. The FCM cluster-based protocol is proposed and evaluated in this chapter. A framework of designing the centralized cluster-based protocols is also proposed. Simulation results are provided for analyzing the systems and the experiment results are given for validation.

Chapter 5 discusses the problem of optimizing the cluster formation and cluster head selection. A cluster-based protocol using HSA is proposed. The performance of HSA when solving the optimization problem for this protocol is investigated and compared with other nature-inspired optimization methods such as GA and PSO in simulation. HSA is later employed to support the operation of the real-time WSNs designed with the framework proposed in Chapter 4. Experiment results are provided to prove the enhancement of the network performance when using HSA.

Chapter 6 presents the optimal data aggregation tree construction problem. The IWD algorithm is introduced and later adopted to solve the optimization problem formed. A modification of the basis IWD is proposed to improve the construction of the the optimal tree. Computational experiments are conducted to study the performance of the methods with the comparison with a similar type optimization method, ACO.

Chapter 7 concludes the research work presented in this thesis and suggests some future works which can be extended further.

Chapter 2

Energy Efficient Routing and Optimization Methods in Wireless Sensor Networks

2.1 Introduction

The main objectives in operating WSNs are to have efficient and reliable communication, to achieve low energy consumption and to prolong the network lifetime. In such a resource constraint system like the WSN, the energy conservation is the main concern for network development. In general, the energy consumption of a WSN as well as a sensor node is distributed into three main tasks: *sensing, data processing* and *communication*. Since the *communication task* of the WSNs usually requires significant amount of energy compared to the other tasks [6], it is essential to design energy efficient communication protocols to obtain optimal energy usage. A number of attempts for enhancing the WSNs energy efficiency have been carried out at different layers of the network protocols.

As described in Chapter 1, the communication protocol stack of WSN normally includes four layers: Physical (PHY) layer, Medium Access Control (MAC) layer, Network (NWK) or routing layer and application layer. The implementation of application layer needs to follow the requirements of specific applications. Meanwhile the low layers can be designed in a generic way that can be employed in various applications.

The PHY layer implements the communication hardware for the WSNs that takes the responsibility of signal transmission and reception. The functions of PHY layer include selecting the transmission frequency, generating carrier frequency, detecting incoming signal, performing modulation scheme and encrypting data for security requirement [6]. For WSN, energy savings starts at the PHY layer that is related to minimize the energy consumption for running radio circuitry and bit streaming transmission. Recent research studies include physical-layer requirements, low power radio design, power-aware transmission schemes and modulation schemes. Some of the research works about these aspects have been proposed in [35–40].

The MAC layer is to manage the communication between directly connected sensor nodes as introduced in Chapter 1. This layer is responsible for the multiplexing of data streams, data frame detection, medium access and error control [5]. It is required the MAC layer to obtain high throughput and low error rates. However, the design of the MAC protocol in a WSN is subject to various constraints such as energy, topology, and network changes. Minimizing energy to extend the network lifetime is its primary goal. Energy wastage in the network needs to be avoided, it can be caused due to packet collisions, overhearing, excessive re-transmission, idle listening and protocol overheads [41]. The development of MAC protocol should also consider the trade-off between reliable network control and energy efficiency. Some protocols developed to conserve energy are based on low duty cycle with active and sleep modes such as Sensor-MAC (S-MAC) [42], Timeout-MAC (TMAC) [43], DMAC [44], etc. Other protocols reduce energy wastage by attempting to avoid collision while transmitting packets. These protocols can be categorized into either schedule-based protocols or contention-based. In schedule-based protocols, only one sensor nodes in the neighborhood of the receiver can send data packet at a time. Time Division Multiple Access (TDMA) or a synchronization mechanism is often used in this protocol to allocate the transmission time for each sensor node. Some of the typical schedule-based protocols are TRaffic-Adaptive Medium Access protocol (TRAMA) [45], Self-Organizing Medium Access Control for Sensor Networks (SMACS) [46], LEACH [47] etc. The contention-based protocols provide a transmit opportunity for all the nodes to transmit data packet to the receiver [41]. Mechanisms such as Carrier Sensing Multiple Access (CSMA) that utilize frequent channel sampling for detecting possible transmission are employed in this type of protocols. Power Aware Multi-access protocol with Signaling (PAMAS) [48], Sparse Topology and Energy Management (STEM) [49], Berkeley Media Access Control (B-MAC) [50] and the improvements of B-MAC such as Speck Media Access Control (SpeckMAC) [51] and Zebra MAC (Z-MAC) [52] are some of these protocols. Among these protocols, B-MAC has been implemented practically on Crossbow sensor nodes.

The PHY layer and MAC layer for low-rate WPANs have been standardized in the IEEE 802.15.4 [15]. The services offered by IEEE 802.15.4 have been employed in many WSN hardware platforms such as TelosB [53], Mica Mote family

[54], IRIS [55], Tmotes Sky [56], EYES node [57], ProSperkz [58], etc. However, IEEE 802.15.4 covers only the two lower layers of the communication protocols and leave network layer for developers so that it can be designed to adapt specific applications. As PHY and MAC layer, the network or routing layer for WSNs is required to be energy efficient. The routing layer is built on top of the MAC layer and is responsible for delivering data packets within the network from the sources to the destinations. The methods of constructing the route and maintaining the connectivity of sensor nodes have a great influence on the energy consumption of the WSNs. The next section discusses about technical challenges to develop routing protocols for WSNs. Some typical routing protocols proposed in the literature and their limitation are also presented.

2.2 Routing Protocols for Wireless Sensor Networks

2.2.1 Overview of routing protocols in WSNs

The routing protocols for WSNs have their own challenges which are different from the traditional protocols for wired networks or other wireless networks such as cellular networks.

- **Data-centric communication**: WSNs normally employs data-centric communications that are based on data measurement without caring about the exact node identification. Meanwhile, the traditional networks use the node-centric communication where nodes exchange data with each other by using unique addressing.

- **Resource constraints**: The sensor nodes have a very limited resources, thus the routing method cannot be designed with heavy computational burden.

- **Network lifetime**: Network lifetime is always an important requirement when planning the operation of WSNs. Many applications require long-term deployment of the WSNs without any direct access of users. Proper selection of the routes to deliver data packets can result in extension of network lifetime. However, there are many aspects need to be considered when choosing a efficient route. The routing protocols must keep the number of control packets for route establishment to be in minimum volume. They should not only minimize the route maintenance cost but also transfer data packets via the most energy-efficient routes.

- **Scalability**: WSNs consists of a number of sensor nodes. The sensor nodes may leave the network at any time due to their failure or energy depleted, and rejoin when they get recovered.

- **Redundancy minimization**: Usually the sensor nodes are densely deployed in an area of interest. It has high chance that the sensing values measured by the sensor nodes in their mutual neighborhood are tightly correlated. Therefore, data sent to the destination might be redundant that causes inefficiency of energy usage.

- **Application-specific behaviour**: Since the WSNs are highly application-specific, the design and configuration of the routing protocols need to be change to adapt different application.

2.2.2 Classification and Operation of routing protocols in WSNs

In order to find an energy efficient and reliable route for data delivery, the routing protocols for WSNs proposed in the literature have taken several strategies to achieve energy savings such as data aggregation, in-network processing, clustering, etc. Among those, data aggregation is a potential technique that can improve the energy efficiency of the WSNs significantly. Data aggregation is defined as the process of aggregating the data generated from different source nodes to eliminate the redundant data packets transferred and provide the compact information to the BS [19]. Only the most critical data from the sensor nodes collected and delivered to the destination enable energy savings for the operation of the WSNs. Data aggregation can be integrated in various routing protocols in different manner depending on the structure of the WSNs established by the protocols. Routing protocols in WSNs can be classified into flat-based routing protocols, cluster-based routing protocol and location-based routing protocol based on the network structures [111]. The following section introduces about each categories and the application of the data aggregation techniques in routing protocols.

a) Flat-based routing protocols

In flat network, all of the sensor nodes play the same role and incorporate each other to perform sensing task. For example, flooding and gossiping [59] are two classical protocols that transfer the data in WSNs without the need of any routing algorithms and topology algorithm. Flooding is the simplest way of sending information from the source to the destination. Each sensor receiving a data packet broadcasts it to all its neighbors, the process keeps going on until the packets reach the destination or it has traveled through the maximum number of hops. The main problems of flooding are *implosion* caused by duplicated messages sent to the same node, *overlap* when two nodes sensing the same region and send similar packet to the same nodes, and *resource blindness* by consuming large a mount of energy without considering energy constraints [111]. Gossiping is an improved version of flooding where the receiver node sends the packet to random selected neighbor. The implosion is avoided but the number of hops may be very large, thus this causes delay in propagation of data through the nodes. Most of these drawbacks have been solved by other advanced flat-based routing protocols those have proposed in the literature and thus the performance of the networks is much further enhanced and optimized. Some of these protocols are Sensor Protocols for Information via Negotiation (SPIN) [60], Directed Diffusion (DD) [61], Energy-aware Routing (EAR) [62], etc.

SPIN [60] is a routing protocol that employs negotiation mechanism in which each node advertises new data to its neighbors and interested neighbors acquire the data by sending a request message. SPIN achieves energy conservation by sending data that describe the sensor data instead of sending all data. However, the SPIN's data advertisement cannot guarantee the delivery of data because if the interested nodes are far away from the source nodes and the nodes between them are not interested in the data, such data never reach to the destination.

DD [61] is a data-centric routing protocol which diffuse data through sensor nodes by using a naming scheme for the data so that unnecessary operations of network layer routing is avoided in order to save energy. However, DD cannot be applied to all WSNs applications since it is based on a query-driven data delivery model [111].

EAR [62] is concerned with increasing the lifetime of the network by using a set of sub-optimal paths to route the data. A probability function, which incorporates with energy consumption of each path, is used to chose these paths.

b) Cluster-based routing protocols

For the advantages of scalability and efficient communication, cluster-based protocols attract a great attention in WSNs research. The concept of grouping the sensor nodes into clusters and rotating CH roles among cluster member enable to balance data traffic load and energy consumption in the networks. Data aggregation and fusion at the CHs assist to reduce the amount of information transferred in the network. Transmission power of the member nodes within the clusters can be decreased. Thus, energy saving is achieved. For all these reasons, cluster-based protocols is able to be employed for a wide range of applications such as patient health care monitoring, machinery condition monitoring, building management, security surveillance which are the main objectives of this research. In the literature, there are many studies that attempt to obtain efficient performance of cluster-based WSNs. Each of these studies has different approaches to optimize WSNs operation.

LEACH [18] is a typical cluster-based protocol using a distributed clustering formation algorithm. The cluster heads are selected with a predetermined probability, other nodes choose the nearest cluster to join, basing on the strength of the advertisement message they received from the cluster heads. After forming the clusters, cluster heads compress data arriving from the sensor nodes and send an aggregated packet to the BSs in order to reduce the amount of information sent to the BSs. However, the distributed mechanism of forming clusters in LEACH may result in poor cluster structure in which some clusters contain a much larger number of nodes than others do. Furthermore, randomly choosing CHs lead to the fact that there is a high chance for one node to be a CH during several rounds of operation, this node consumes more energy and may die out within a short period of time. LEACH-C [47] is an improved version of LEACH which uses centralized mechanism to form clusters. The base station has the responsibility to select the CHs by using simulated annealing algorithm and to allocate other nodes to a particular cluster.

Other improvements of LEACH have been proposed in [63], [64]. These methods improve the efficiency of the cluster by preventing neighbor nodes from becoming CHs in the same round. The balanced clustering approach takes into account virtual partition which is calculated by mathematical approximation of the regional residual energy.

Some clustering approaches utilize location information without considering the energy model. Shin et al. [65] and Chan et al. [66] proposed an optimizing algorithm for clustering formation in WSNs; however only location data of all the sensor nodes are used, and there is no energy model considered.

Power-efficient Gathering in Sensor Information Systems (PEGASIS) [67] is an improvement of LEACH protocols. Instead of forming multi clusters as LEACH, PEGASIS forms chains from sensor nodes so that each node transmits and receives from a neighbor and only one node is selected from that chain to transmit to the base station. PEGASIS achieves better performance due to eliminating overhead caused by dynamic cluster formation in LEACH. However PEGASIS introduces excessive delay for the node further from the sink, and the single node gathering data can become a bottleneck.

Threshold sensitive Energy Efficient sensor Network (TEEN) [68] and its extension version Adaptive Threshold sensitive Energy Efficient sensor Network (APTEEN) [69] are hierarchical protocols designed to respond to sudden changes in the sensed attributes. APTEEN is enhanced to capture periodic data collection. In these protocols, sensor nodes transmission is trigger by using thresholds of sensed attribute which broadcasted by the cluster head. The main drawbacks of the two approaches are the overhead and complexity of forming clusters in multiple levels, implementing threshold-based functions and dealing with attribute-based naming queries. Furthermore, these protocols do not attempt to optimize the formation of the clusters but improve the performance of the networks by enhancing data aggregation and fusion.

Liu et al. [70] introduced a hierarchical clustering approach by proposing the re-clustering strategy and a modified redirection scheme to increase the lifetime of a network. This approach forms hierarchical clusters and attempts to achieve an equality of density in each cluster.

c) Location-based routing protocols

In this kind of routing, sensor nodes are addressed by means of their locations. The distance between neighboring nodes can be estimated based on incoming signal strength. Relative coordinates can be obtained by exchanging such information between neighbors. Alternatively, the location of the sensor nodes may be identified by communicating with a satellite by using Global Positioning System (GPS). Some of such protocols are GAF [71] and GEAR [72].

2.3 Optimization Methods for Energy Efficient Routing Protocols

2.3.1 Clustering Algorithms for cluster-based protocols

Although clustering techniques are commonly known to be used for data management and pattern recognition, they can be applied in a large number of problems in different research areas. These techniques as mentioned above provides a method to reduce the size of the network to be controlled by grouping sensor nodes into clusters which have manageable size and assists the improvement of energy efficiency [16]. The clustering techniques deal with grouping objects according to a given measure. It emerges a high degree of structure for an unstructured set of objects. Many clustering algorithms have been proposed in the literature. However, the main problem with the data clustering algorithms is that it cannot be standardised. Algorithms developed may give best result with one type of data set but may fail or give poor result with data set of other type [73]. Clustering algorithms can be broadly classified into two categories: (1) Parametric Clustering and (2) Non-Parametric Clustering [74]. The parametric clustering methods attempt to minimize a cost function or an optimality criterion, which associates a cost to each instance-cluster assignment. This type of method usually includes some assumptions about the underlying data structure; it is assumed that set of parameters is finite. K-Means, Fuzzy C-Means, Gaussian Mixture Density Decomposition, etc., are classified into parametric clustering algorithms [75, 77, 78]. Meanwhile, non-parametric type assumes that the data distribution cannot be defined in terms of such a finite set of parameters. They can often be defined by assuming an infinite dimensional. No assumption about the underlying data distribution is required. Furthermore, they do not require an explicit representation of the data in a Euclidean form. They only require a matrix with the pair-wise similarities based on a predefined distance. However, such a matrix containing the pair-wise similarities sometimes can require a lot of storage space, thus making the algorithm inapplicable in much of real life problems, where the data set to cluster is typically large. Density-based Spatial Clustering of Applications with Noise (DBSCAN) and hierarchical clustering algorithms, which includes agglomerative methods like BIRCH, CURE, ROCK, etc., and divisive methods like DIANA and MONA, belong to this type [79]. Table 2.1 shows the comparison of some typical clustering algorithms.

Table 2.1: Typical clustering algorithms

Clustering algorithms	K-means	Fuzzy C-means	Gaussian Mixture Density Decomposition	Hierarchical clustering algorithms	Density-based Spatial Clustering of Applications with Noise (DBSCAN)
Advantages	- It is fast, robust and easier to understand - It is relatively efficient. - It gives best result when data set are distinct or well separated from each other.	- It gives best result for overlapped data set and comparatively better then k-means algorithm. - Unlike k-means, here data point may belong to more than one cluster.	- It gives extremely useful result for the real world data set.	- No a-priori information about the number of clusters is required. - Easy to implement and gives best result in some cases.	- It does not require a-priori specification of number of clusters. - It is able to identify noise data while clustering. - It is able to find arbitrarily size and arbitrarily shaped clusters.
Disadvantages	- The learning algorithm requires a-priori specification of the number of clusters. - The learning algorithm provides the local optima of the squared error function. - If there are two highly overlapping data, k-means will not be able to resolve that there are two clusters.	- A-priori specification of the number of clusters is required. - Euclidean distance measures can unequally weight underlying factors.	- Algorithm is highly complex in nature.	- The algorithms are sensitive to noise and outliers, break large clusters, and are difficult to handle different sized clusters and convex shapes. - No objective function is directly minimized - Sometimes, it is difficult to identify the correct number of clusters by the dendogram. - It is more complex that parametric-method.	- It fails in case of varying density clusters. - It is more complex that parametric methods.
Complexity	$O(NKd)$	$O(N)$	-	$O(N^2 \log N)$	$O(N \log N)$

In cluster-based WSNs, clustering techniques are essential for network formation, which defines the structure of the network and the connectivity among the sensor nodes for data transmission. However, WSNs and their fundamental components, the sensor nodes, have very limited ability of computation, small memory storage and finite energy sources. Therefore, the clustering algorithms used for cluster formation should be simple, fast and efficient. Amongst the algorithms mentioned above, K-Means and Fuzzy C-Means are good candidates for improving the cluster formation of cluster-based WSNs.

K-means is one of the most well-known and simplest clustering techniques [75, 76]. The objective of k-means is to organize a given number of objects into k disjunct groups in a hard partitioning manner. The main idea is to define k centroids - one for each cluster. Based on the initial selection of k centroids, k-means iteratively updates the clustering until the algorithm converges, i.e a stopping criteria such as maximum number of iterations or no change of fitness value is satisfied. In every iteration, each data point (object) is associated with the nearest centroid. When all points are allocated into clusters, the present iteration is finished. The algorithm recalculated k centroids based on the network configuration obtained in the present iteration. After that, the next iteration is started to improve the clustering. This iterative process will update the centroids and the associated clusters in each step until no further improvement can be achieved.

k-means has already been successfully used in the context of wireless ad-hoc network by Fernandess and Malkhi [20] in order to limit the amount of routing information stored and maintain at individual hosts. In WSNs, Tan et. al. [21] have employed k-means in their proposed BPK-means protocol for assistance of the network formation.

FCM is another well-known fuzzy clustering algorithm that was first proposed by Bezdek [77]. Like k-means, the objective of the FCM clustering algorithm is also to group a set of objects into a given number of clusters. However, it is different from k-means that instead of hard partitioning data points (objects) into only one particular cluster, FCM establishes overlapping clusters of the data points. Each object is assigned a degree of belonging to each cluster created rather than completely being a member of just one clusters. This algorithm is also potential to be applied in WSNs that have high degree of randomness.

2.3.2 Nature-inspired Optimization Methods

There are many nature-inspired optimization methods proposed in the literature [80, 81]. The optimization methods can be classified into static optimization and dynamic optimization. Static optimization deals with minimizing or maximizing a quantity for a given instant of time, while dynamic optimization refers to the process of minimizing or maximizing the cost/benefits of some objective function over a period of time [82]. In this study, the problem formulation demands static optimization, since optimization process is performed periodically over the operation of the WSNs, at each instance of which the network is temporally static.

(a) Optimization Methods for Cluster-based Network

In cluster-based WSNs, traditional clustering algorithms like k-means or FCM are used to cluster the network based on the distance measure. In order to achieve higher efficiency, other parameters like energy need to be considered in the objective function. The more complex problems require optimization methods that can solve the problems in a proficient manner and achieve global or near global optimum.

Evolutionary algorithms eradicate some of the above mentioned difficulties and are quickly replacing the classical methods in solving practical problems [83–85]. Evolutionary algorithms typically intend to find a good solution to an optimization problem by trial-and-error in a reasonable amount of computing time. The most prominent evolutionary algorithm is GA which is based on natural genetics. GA was developed by John Holland in the 1960s and 1970s [86]. The essence of genetic algorithms involves the encoding of an optimization function into arrays of binary or character strings to represent chromosomes. The population of GA consists of various set of chromosomes. The initial population then evolves by generating new generation individuals via crossover of two randomly selected parent chromosomes, and the mutation of some random bits. Whether a new individual or offspring is selected or not is based on its fitness value, which is determined from the objective function [86, 87]. Later, inspired by the swarm behavior of fish and bird schooling in nature, PSO was developed in 1995 by James Kennedy and Russell C. Eberhart [88]. In PSO, each single solution is represents a particle in the search space. The particles fly through the problem space by following the current optimum particles. During flight, each particle adjusts its position according to its own experience, and the experience of neighboring particles, making use of the best position encountered by itself and its neighbors. The swarm direction of a particle is defined by the set of particles neighboring the particle and its history experience [88, 89]. In WSNs, GA and PSO have been applied to solve several optimization problems. Konstantinidis et al. applied GA to solve a problem of maximizing the coverage and lifetime of the WSN [91]. The coverage optimization problems of the WSNs are also formulated and solved by using GA in [92, 93]. The authors in [94] adopted PSO to create an optimal power allocation scheme for the WSN to achieve energy savings and reliable communication.

In [31], Latiff et al. applied evolutionary algorithms to obtain the optimization of CHs selection based on an objective function which contains the residual energy and the relative distance between normal nodes and CHs. The results obtained by using GA and PSO have been compared with and proved to outperform that achieved by some conventional methods.

However, GA inherently suffer from high computational complexity. The most popular versions of GA use binary encoding and they consume huge memory for achieving high precision. PSO consumes multiple banks of memory to accommodate the particles and their personal bests. In this study, real-time operation is desired and lighter algorithms are preferred. HSA requires less effort of tuning parameters and is easy to implement. Furthermore, its diversification and intensification are well-controlled [95]. Hence, computational payload may reduced and HSA would be appropriate to support the operation of WSNs.

(b) Optimization Methods for Tree-based Network

Additionally, in the *tree-based protocols*, sensor nodes are organized into a tree where data aggregation is performed at intermediate nodes, which are located at the junctions of tree branches. The aggregated data packets are later routed to the root node, i. e. the BSs. The tree-based protocols are suitable for applications, that involve in-network data aggregation [19]. Examples of the applications can be forest fire detection, safety monitoring in industrial plants, etc., where the measurement provides the most useful information about the safety conditions. One of the main objectives of the tree-based protocols is to optimize the construction of a data aggregation tree in terms of energy efficiency. This optimal aggregation tree is recognized as NP-Hard [96], which is equivalent to Steiner tree, weighted set cover problem [97].

Greedy Incremental Tree (GIT) is an approximation algorithm for finding optimal aggregation trees based on DD, that is proposed to establish an energy-efficient path in [98]. Krishnamachari et al. compare Shortest Path Tree (SPT), Center at the Nearest Source (CNS) and a modified version of GIT to manifest the advantage of data aggregation and to demonstrate enhanced methods for constructing the aggregation tree [97].

Misra, R. et al. apply the ant colony algorithm to solve the data aggregation problem [99]. The ACO [100, 101] proposed by Dorigo et al., is a well-know swarm intelligent approach which mimics the foraging behaviour of the ant society. These ants deposit pheromone on the ground in order to mark some favorable path that should be followed by the other colony's member. This approach is utilized in [99] to determine the optimal paths from different sources to the BSs, data packets are aggregated at the nearest relay nodes to the sources. Liao et al. proposed an improvement of achieving optimal data aggregation tree in WSNs using ACO by extending the search region around the routing paths, and thus increasing the probability of finding aggregation nodes [34]. These works have shown to outperform the conventional methods in terms of energy conservation.

2.4 Conclusions

This chapter presents a brief summary of the routing protocols proposed in the literature. The design of routing protocols for WSNs is a challenging task that needs to take into accounts the energy efficiency, robustness and scalability. In the resource constraint systems like WSNs, the novel optimization methods inspired by nature processes offers algorithmic design principles that are suitable to plan and operate the WSNs in an efficient way. Some of issues of routing protocol development have been indicated such as the inefficient network construction or poor selection of the aggregation nodes for gathering data that result in energy wastage and shorten the network lifetime. These issues are also focuses of this thesis that are discussed in detailed and solved in the following chapters.

Chapter 3

WSNs Systems Model and Analysis

3.1 Introduction

This chapter presents the system model and analysis of some mechanisms that can be used to support energy conservation of the WSNs.

3.2 Wireless Sensor Nodes' Energy Model and Cluster Head Rotation for Balancing Energy

3.2.1 Energy Model of the Wireless Sensor Node

In a WSN, sensor nodes consume energy to receive data from other nodes, process or aggregate these data and transmit to the other nodes. Sensor node components and associated energy consumption for communication are shown in Fig. 3.1 where E_{Tx} and E_{Rx} are transmitting and receiving energy consumption. When data aggregation mechanism is applied, the node's energy consumption that is used to fulfill data aggregation task is denoted as E_{da}. Both the free space and multipath

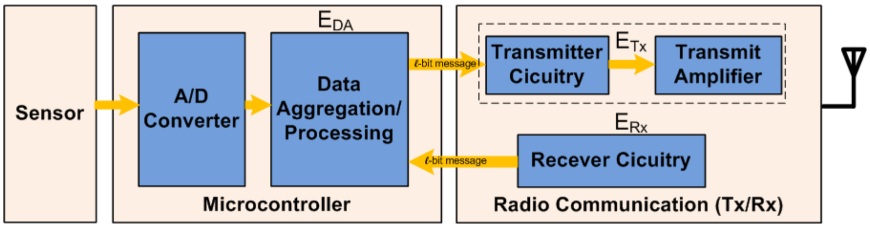


Figure 3.1: Sensor Node Components and Energy Consumption for Data Aggregation and Communication Task

fading channel models in [47] are used to compute energy dissipated during the process of transmitting and receiving information. The energy consumption for transmitting a *l-bit* message over a distance d, in m, is

$$E_{Tx} = E_{elec} * l + E_{fs} * l * d^2,\ d < d_0 \tag{3.1}$$

$$E_{Tx} = E_{elec} * l + E_{mp} * l * d^4,\ d \geq d_0 \tag{3.2}$$

and for receiving this message respectively is:

$$E_{Rx} = E_{elec} * l \tag{3.3}$$

where E_{elec} is the energy spent to operate the transceiver circuit, i.e. transmitter and receiver, E_{fs} and E_{mp} are the energy consumption of transmitting one bit data to achieve an acceptable bit error rate and is dependent on the distance of transmission in the case of free space model and multipath fading model. If the transmission distance is less than a threshold d_0, the free space model is applied; otherwise, we use the multipath model. The threshold d_0 is calculated as

$$d_0 = \sqrt{E_{fs}/E_{mp}} \tag{3.4}$$

3.2.2 Cluster Head Rotation for Balancing Energy in Wireless Sensor Nodes

Considering a small scale WSN such as a WBAN with n nodes, these n nodes are organized in one group or cluster that is deployed on a human body in the case of WBAN and is stationary on the body during the operation. One of the nodes is chosen as the gateway or CH that gathers data from all other nodes and sends the concise data to a BS. The energy consumption of each node to transmit l-bit messages to this gateway is given by using Equation (3.1)

$$E_{node} = E_{Tx}(l, d) = E_{elec} * l + E_{fs} * l * d^2 \tag{3.5}$$

where d is the distance between the node and the gateway. The energy consumption of the gateway to receive data from all nodes and send to a base station is as follows

$$\begin{aligned} E_{gateway} &= E_{Tx}(n * l, d) + E_{Rx}((n-1) * l) && (3.6) \\ &= E_{elec} * n * l + E_{fs} * n * l * d^2 + E_{elec} * (n-1) * l && (3.7) \\ &= l * [(2n-1)E_{elec} + n * E_{fs} * d^2] && (3.8) \end{aligned}$$

To solve the problem of just relying on one energy hungry gateway to communicate with the BS, a selective gateway method to select among the sensor nodes based on their residual energy as the local gateway can be utilized to balance energy consumption of the network. Whenever the residual energy of the gateway reduces below a threshold value, E_{th}, another node among the rest in the WBAN with higher energy level than E_{th} is chosen as the current gateway. After one round of selection when all the nodes' residual energy drops below the threshold, the original threshold value is adjusted lower. It is required to make sure that at least one of the sensor nodes has enough energy supply to become the gateway. The selection process is repeated again as mentioned above until the residual energy stored in all the sensor nodes are used up. The information about residual energy of each individual node is added to the sending data, and is compared at the gateway at each round of data gathering. Decision of changing gateway made by the current gateway is sent to all the other nodes through the acknowledge messages of received data in the next round.

The simulation result shown in Fig. 3.2 illustrates the residual energy of the gateway based on the conventional fixed gateway case and the last node running out of energy when using proposed selective gateway method with 10 nodes deployed in an area of 0.4 m $\times$ 1.7 m, a 200-bit message (i.e. header, payload, metadata, etc.) is transferred from the sensor node to the gateway every round. Assuming that the distance between the gateway and the base station is within 200 m and all nodes have the same initial energy 0.5 J, the gateway in the first method spends all of its energy after 200 rounds of receiving and transmitting data, meanwhile in the second method, every node runs out of energy after an average number of 1700 rounds.

Fig. 3.3 shows the comparison between the lifetime of the single unique gateway and that of the last node which runs out of energy in the selective gateway method with respect to the number of nodes deployed in the WBAN. When the gateway is fixed, increasing number of nodes causes a short life of the gateway and thus shortens the network lifetime. In another case, where the gateway is selected based on the residual energy of the sensor nodes, the average lifetime of each node

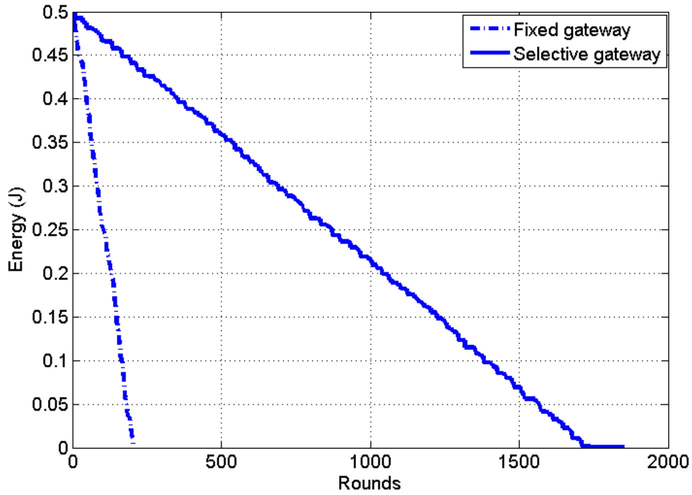


Figure 3.2: Residual energy levels of the conventional fixed gateway and the selective gateway

in the network is much longer and its performance is independent of the number of sensor nodes. Therefore, the network lifetime, which does not depend on a unique local gateway, is much more improved.

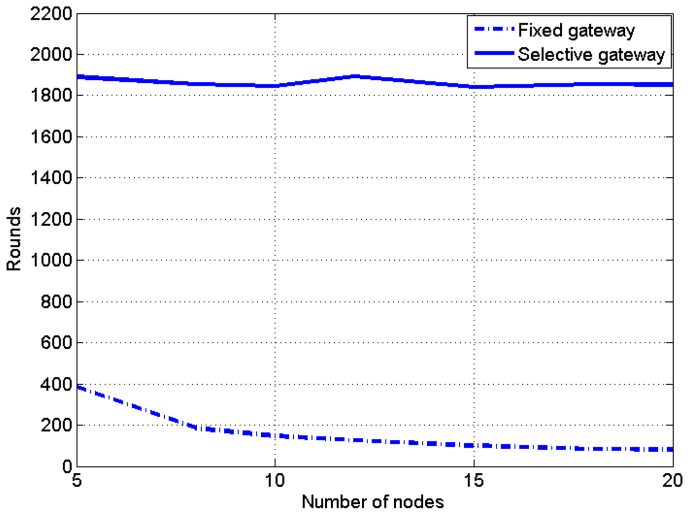


Figure 3.3: Gateway life time with different number of nodes in the network

By changing the gateway based on residual energy, the energy among all nodes is balanced, the time that gateway exists in the network is longer and thus, it guarantees the connection with the base station. Furthermore, when an energy harvesting source is added, the proposed method provides a benefit of utilizing energy scavenged from all of the nodes as well as let the gateway have enough time to recharge and restore its energy capacity and therefore prolong the network lifetime much more.

3.3 A Single Sensor Node Hardware System

3.3.1 Sensor Node Energy Consumption

As described in Chapter 1, a wireless sensor node consists of a sensor module, a microcontroller, a transceiver and a power supply. In general, the energy consumption of a WSN as well as a sensor node is distributed into three main tasks: *sensing, data processing and communication.* The energy consumption of the sensing module is dependent on the nature of the applications which could be either sporadic sensing or constant monitoring. The complexity of the observation also has a great influence on the energy consumption of the sensor. The characteristics of some types of sensor popular used is shown in Table 3.1. It is observed that the current consumption of these sensors is in the range of $0.01\,mA$ to $1\,mA$ considerable small compared with that of other components such as microcontroller or transceiver which can be in the range of tenth mA.

Data processing is another activity that requires energy consumption of the sensor node. In microcontrollers using CMOS technology, the power consumption

Table 3.1: Example current consumption and related characteristics of sensors [102]

Sensor	Accuracy	Interchange -ability	Sample rate [Hz]	Startup [ms]	Current [mA]
Photoresistor	N/A	10%	2000	10	1.235
I2C temperature	1 K	0.20 K	2	500	0.15
Barometric Pressure	1.5 mbar	0.5%	10	500	0.01
Bar. press. temp.	0.8 K	0.25 K	10	500	0.01
Humidity	2 %	3 %	500	500-3000	0.775
Thermopile	3 K	5 %	2000	200	0.17
Thermistor	5 K	10 %	2000	10	0.126

of data processing, (P_p), can be formulated as follows [103]

$$P_p = C_{total}V_{dd}^2 f + I_{leak}V_{dd}(N/f) \tag{3.9}$$

where C_{total} is the total switching capacitance; N is the number of clock cycles taken by the computation; V_{dd} is the supply voltage and f is the switching frequency. The first term in Equation (3.9) is the main power consumption of the component and the second term is the power loss due to leakage current, I_{leak}.

Data communication consumes a remarkable amount of sensor node's energy as this task involves both data transmission and reception. The average power consumption, P_c of the radio component is described as [104]

$$P_c = N_{Tx}[P_{Tx}(T_{on-Tx} + T_{st-Tx}) + P_{out}T_{on-Tx}] + N_{Rx}[P_{Rx}(T_{on-Rx} + T_{st-Rx})] \tag{3.10}$$

where P_{out} is the output transmit power; $P_{Tx,Rx}$ is the power consumption of transmitter/receiver; $N_{Tx,Rx}$ is the average number of times per second that transmitter/receiver is used; $T_{on-Tx,on-Rx}$ is the transmitter/receiver ON time interval and $T_{st-Tx,st-Rx}$ is the transmitter/receiver START-UP time. The ON time interval can be computed as

$$T_{on} = \frac{L}{R} \tag{3.11}$$

where L is the data packet size and R is the data rate.

Thus, in order to minimize P_c, all the parameters listed in Equation (3.10) need to be handled efficiently during communication. For example, power saving can results from the reduction of the transceiver's ON time interval achieved by transferring small size data packet at a high rate. In summary, the general strategies which can be used for sensor node power management are limiting energy wasted on any unnecessary tasks, turning off some parts or the whole circuit when they are not in use and minimize energy consumption to complete tasks. Table 3.2 shows the state of the sensor node using power mode scheduling algorithm [104] which saves energy by defining active and sleep mode of each component to use properly.

Table 3.2: Operation modes of the sensor node's components

States	Processors	Memory	Sensor	Radio
s_0	Active	Active	On	Tx,Rx
s_1	Idle	Sleep	On	Rx
s_2	Sleep	Sleep	On	Rx
s_3	Sleep	Sleep	On	Off
s_4	Sleep	Sleep	Off	Off

Depending on the requirement of the application, the decision of changing from a high power consumption state to a lower power consumption state is made to save more energy. Fig. 3.4 illustrates a scenario in which the sensor node tunes from state s_0 to state s_k consuming less energy and tunes back to state s_0, and hence it saves some energy.

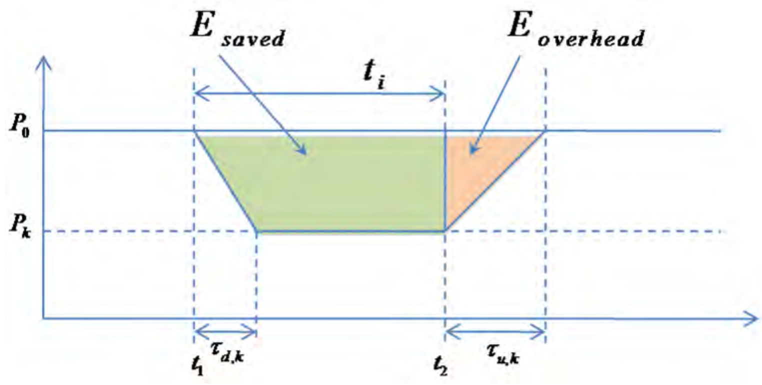


Figure 3.4: Operation modes of a sensor node's main components

At time t_1, the sensor node is operating in state s_0 which has power consumption P_0 and decides to change to state s_k which has power consumption $P_k < P_0$. It takes the sensor an interval $\tau_{d,k}$ to reach state s_k so the energy consumed is $\tau_{d,k}(P_0+P_k)/2$, and then the sensor node spends $t_i-\tau_{d,k}$ time units in this state. If remaining in the state s_0 , the total energy consumption is $E_0 = t_iP_0$. Meanwhile, the energy consumed by the sensor node during the time $t_i - \tau_{d,k}$ in state s_k is $(t_i - \tau_{d,k} - \tau_{u,k})P_k$. Therefore, the total energy saving is

$$\begin{aligned} E_{saved} &= t_iP_0 - \tau_{d,k}(P_0 + P_k)/2 - (t_i - \tau_{d,k})P_k \\ &= t_i(P_0 - P_k) - \tau_{d,k}(P_0 - P_k)/2 \\ &= (t_i - \tau_{d,k}/2)(P_0 - P_k) \end{aligned} \tag{3.12}$$

And the overhead energy added due to the transition time of the sensor node from state s_k to state s_0 is

$$E_{overhead} = \tau_{u,k}(P_0 + P_k)/2 \tag{3.13}$$

Obviously, the transition from high power mode to low power mode is meaningful if the $E_{overhead} < E_{saved}$ or, equivalently if the time interval between two mode changing decisions is large enough

$$t_i > \frac{1}{2}\left(\tau_{d,k} + \frac{P_0 + P_k}{P_0 - P_k}\tau_{u,k}\right) \tag{3.14}$$

In conclusion, careful scheduling of such transitions has been considered from several perspectives which are regarded to application objectives, computation capability of the microcontrollers as well as data communication protocols.

3.3.2 Sensor Node with Thermoelectric Generator

This section presents a simple sensor node with low energy consumption that is powered by an energy harvesting source, a Thermoelectric Generator (TEG). Energy harvesting approach provides sensor node and network unlimited power supply. When incorporating with energy-efficiency communication protocol, energy harvesting source will help to enhance the operation of WSNs whereas the design of sensor node can be optimized in terms of weight, size and power efficiency. Ambient energy sources, which are able to convert to electrical energy, include solar cell, motion and vibration, thermoelectric source, wind, etc.

The application scenario is fall detection of patient in medical healthcare services. Fall detection is very significant to support elderly people's safety in a home-dwelling environment or guarantee patients with disabilities in hospitals to be assisted by informing doctors and nurses timely. Deployment of sensors in a WBAN and some methods to identify fall events are described in [105, 106], where accelerometer sensors mounted on human body or clothes are used for fall detection. The sensing signals are transmitted to a base station to process and send to hospital network and doctors' computers as shown in Fig. 3.5 . In this application thermal energy is one of the potential energy sources in which TEG is used to extract energy from human warmth.

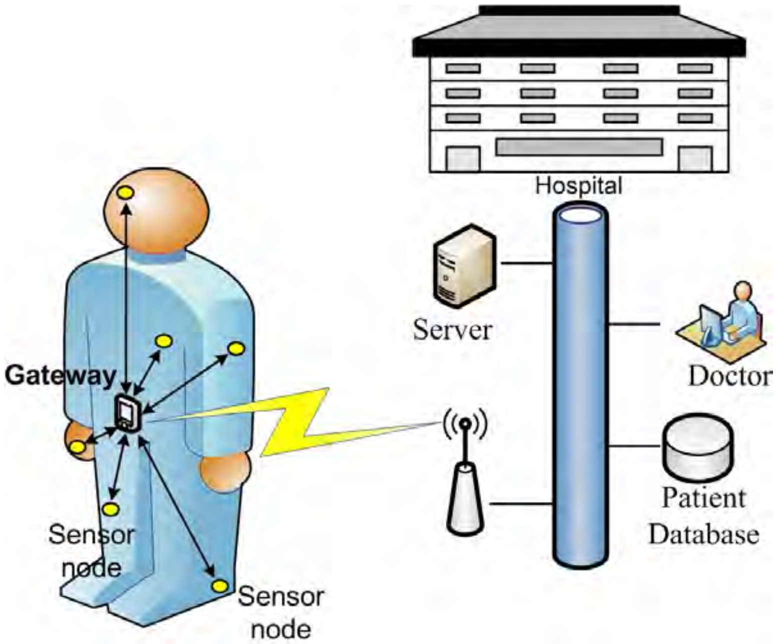


Figure 3.5: Wireless Body Area Network Architecture in Medical Healthcare System

According to Stark [107], the warmness of a human body (and also an animal body) can be used as steady energy source for powering the sensor node in WBAN. The amount of energy released by the metabolism (traditionally measured in Met; 1 Met = 58.15 W/m^2 of body surface) mainly depends on the amount of muscular activity. A normal adult has a surface area in average of 1.7 m^2, so that such a person in thermal comfort with an activity level of 1 Met will have a heat loss of about 100 W. The metabolism can range from 0.8 Met (46 W/m^2) while sleeping up to about 9.5 Met (550 W/m^2) during sports activities as running with 15 km/h. A Met rate commonly used is 1.2 (70 W/m^2), corresponding to normal work when sitting in an office, which leads to a person's power dissipation of about 119 W, burning about 10.3 MJ a day. Once the input thermal energy is known, the equivalent electrical circuit of the Thermal Energy Harvesting (TEH) structure with the TEG, given in Fig. 3.6, is analyzed.

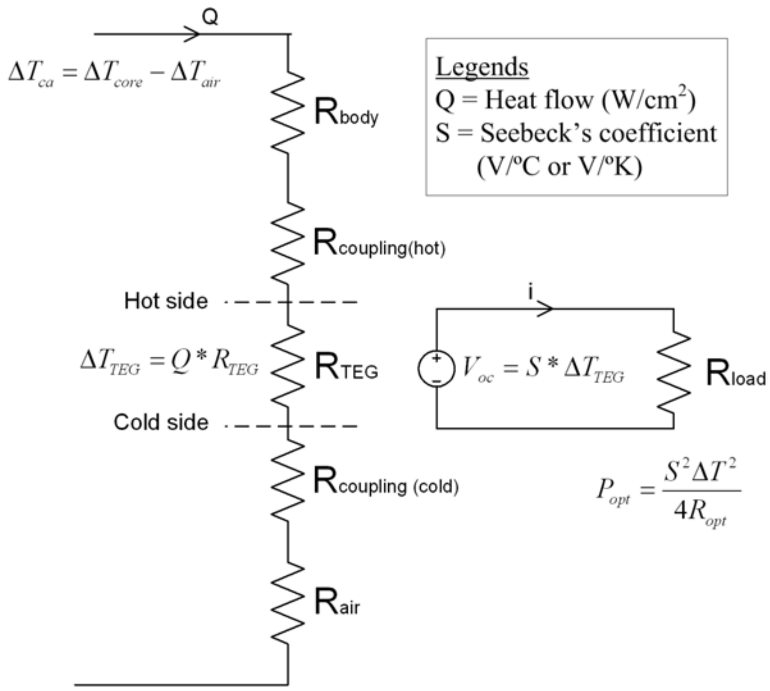


Figure 3.6: Thermal analysis of the thermoelectric generator (TEG)

On the left hand side of Fig. 3.6, it shows the thermal equivalent circuit representation of a TEG in contact with the human skin. The heat flow, Q, takes place in between the body with a core temperature, T_{core}, and the ambient air, T_{air}, with lower temperature through the following thermal resistances representing the body, R_{body}, the interface between body and TEG (hot side), $R_{coupling(hot)}$, the TEG, R_{TEG}, the interface between TEG (cold side), $R_{coupling(cold)}$, and the surrounding air, R_{air}. This results in the Equation (3.15). It describes the relationship among the temperature difference between the body and the air, ΔT_{ca}, the heat flow and the various thermal resistances of the TEH structure. Knowing this relationship, the important factors that affect the overall system efficiency can be identified for improving its performance.

$$Q = \frac{\Delta T_{ca}}{R_{body} + R_{coupling(hot)} + R_{TEG} + R_{coupling(cold)} + R_{air}} \tag{3.15}$$

By keeping the $\frac{R_{TEG}}{R_{body}+R_{coupling(hot)}+R_{coupling(cold)}+R_{air}}$ ratio, ΔT_{ca} and Q as large as possible, the better performance of the overall thermal energy harvesting system is achieved. When there is a temperature difference across the thermoelectric generator structure, the heat resistances, residing in the thermal energy harvesting system, generate certain amount of heat energy loss. These thermal resistances are due to the mechanical structure used to contain the TEG. When heat flow from the body to the TEG and from TEG to air, the material used and the design of the mechanical structure greatly affect the performance of the system. These critical factors are taken into considerations during the design and development of the TEH system.

The concept of TEH is not new to most people and is based on one of the thermodynamic concepts known as Seebeck's theory. Seebeck's effect states that when there is a temperature difference across two dissimilar materials, electric voltage is generated. TEG which is based on Seebeck's theory, is used as the energy converter to transform the thermal energy into electrical energy. In our design, thermoelectric generator is fabricated using aluminum and teflon. Aluminum is used to act as the hot plate designed with a small surface area in order to collect heat fast and cold plate designed in a shape to act as a good heat diffuser. Teflon is used as the insulator sandwiched between the hot and the cold plate so as to effectively reduce the convection and radiation of heat from the hot plate and the cold plate, preventing it from warming up which is highly undesirable as it reduces the thermal gradient between the plates thus affecting the heat flow and power output. The prototype design of the TEH structure with the TEG is shown in Fig. 3.7.

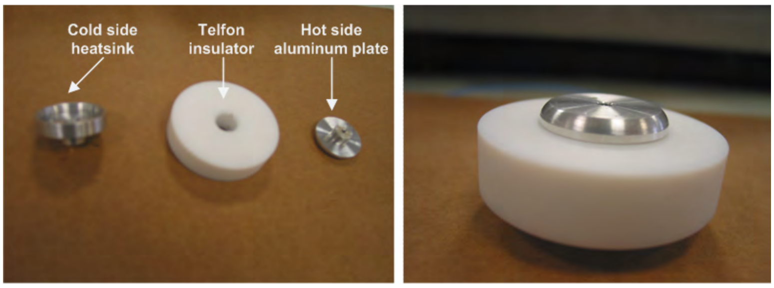


Figure 3.7: Prototype of the thermoelectric generator (TEG)

The power management circuit of the TEH system contains an energy storage and supply circuit, as described by [108], and a voltage regulator circuit illustrated in Fig. 3.8. The operation of the power management circuit is as follows: The electrical energy power harvested from the TEG is stored within a capacitor to a level sufficient to power the loads. The process of storing and releasing energy is controlled by the supply circuit with 2 MOSFET switches, i.e. $Q1$ and $Q2$. During the time when the capacitor is being charged, $Q1$ and $Q2$ are turned off to isolate the TEG source and the radio frequency load. When the built up voltage across the capacitor reaches the preset voltage of 4.9 V, $Q1$ is turned on and then in turn activate the control switch $Q2$. The energy accumulated in the capacitor is then discharged and fed to the voltage regulator. The voltage regulator steps down the input voltage of 4.9 V to 3.3 V to supply to the connected load for its sensing and communicating operation. The thermal energy harvesting system is designed to power a fall detection system. The body sensor node is designed and implemented to be mounted on human body to detect for any falling event. If the falling event is detected, the signal is sent via the wireless communication system to the base station, which is post processed and forwarded to the doctors or nurses for mon-

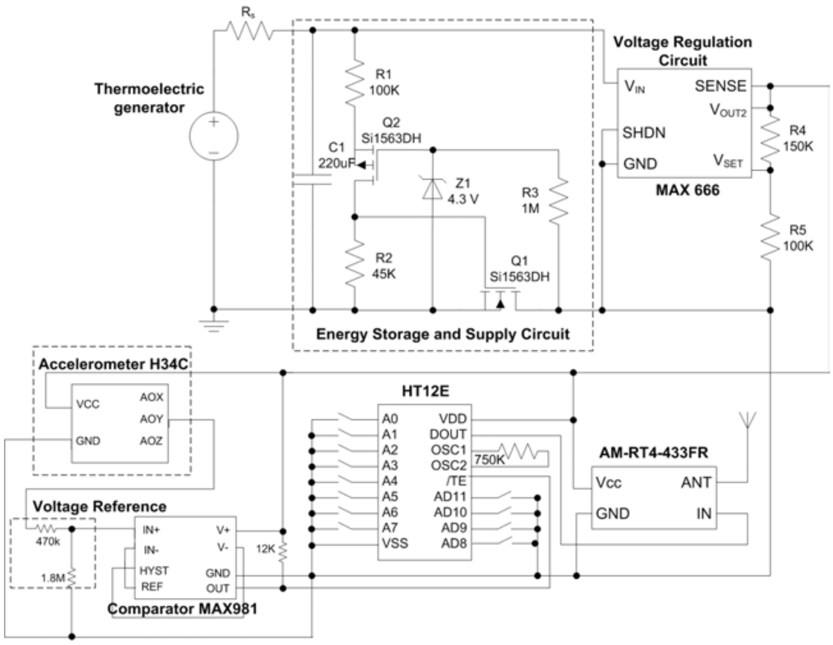


Figure 3.8: Schematic diagram of thermal energy harvesting sensor node for fall detection

itoring of the patient's status or taking some timely responses like activation of emergency ambulance. In this work, the fall detection event is sensed by using an accelerometer. Based on the application requirements, the accelerometer chosen must be able to sense and differentiate, via its internal nomenclature, between an uprightstanding posture and a fallen posture, and subsequently give different output voltage levels to signify the sensed information accordingly. The accelerometer, H34C, obtained from Hitachi, is small in size, high sensitivity in 3-axes i.e. X,Y and Z-axis, very low power consumption of 1 mW @3.3 V supply and is capable of sensing both dynamic and static (tilt) acceleration. In this research, the static sensing mode is used to indicate the body posture, stand or fallen position shown as illustrated in Fig. 3.9. The output voltage (Y) is assigned as the indicated signal for detecting fall. The design principle revolved around the fact that Y is always at its preset voltage level of 1.833 V in a "stand" posture, and always at its $\frac{V_{ref}}{2}$ level in a "fall" posture.

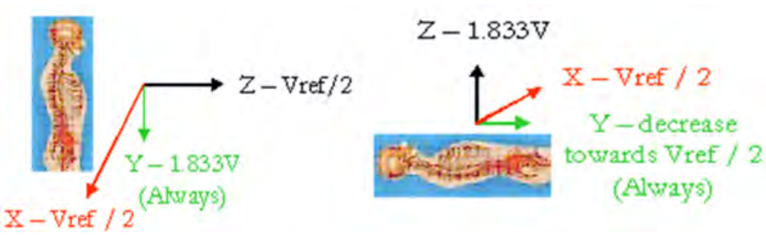


Figure 3.9: Sensing of Body Posture (Stand and Fall) using H34C

To make a comparison between a stand and fall condition, a low power comparator is utilized. The output voltage, V_{out}, of the comparator is determined by the following conditions:

$$V_{in} \quad < \quad (V^{-} + 1.182V), V_{out} = 0V \tag{3.16}$$

$$V_{in} \quad >= \quad 1.182V, V_{out} = V^{+} \tag{3.17}$$

Based on the above mentioned sensing conditions, the signal conditioning circuit to process the accelerometer signal voltage and the comparator output voltage are illustrated in Fig. 3.10.

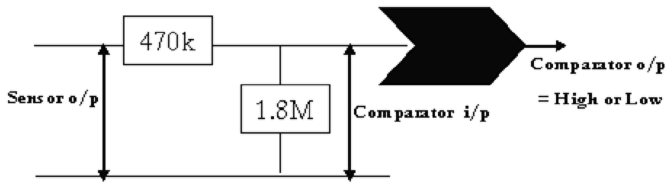


Figure 3.10: Voltage adaptation circuitry for calibrating accelerometer output voltage

As mentioned above, radio transmission consumes the most amount of energy among various components of the sensor node, hence a low power transmitter-receiver pair i.e. AM RT4-433 and AM HRR30-433, which consumes around 10 mW @3.3 V, has been chosen. A matching encoder, HT12E, with very low power consumption, is chosen to encode the communication address and patient identification data for transmission. In the scenario when the patient has fallen, comparator output becomes 0V, therefore enabling transmission of the communicating address and the patient information via the transmitter to relay the encoded bits to the receiver in a wireless manner. The received signal is then decoded by HT12D to alarm for an emergency call of a patient, recognized by his/her identification number.

Various experiments were conducted to verify the proposed TEH system to power the sensor node in a WBAN. The TEH structure is first characterized to determine the amount of harvested electrical power. When different thermal gradients between 3^{o}C to 15^{o}C are applied across the thermal energy harvester, the maximum electrical power generated is illustrated in Fig. 3.11, ranging from 40 μW to 520 μW respectively at the same load resistance of 16KΩ.

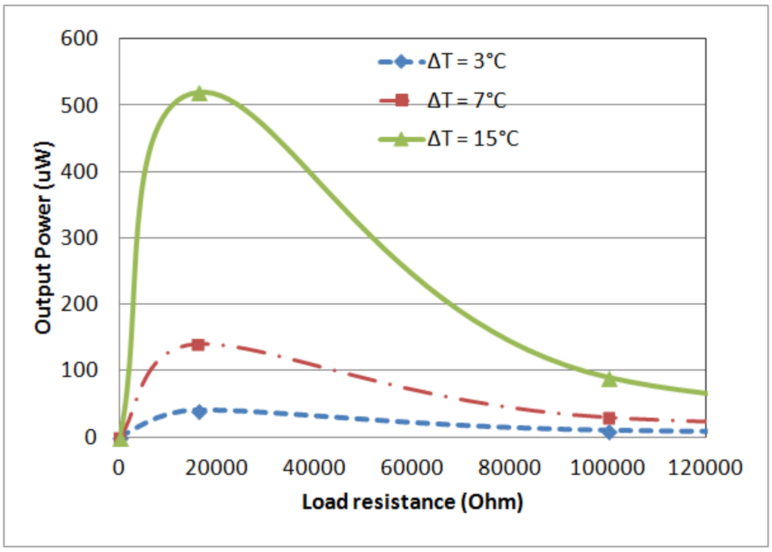


Figure 3.11: Power generated by TEG for various loading conditions

The harvested electrical power of 520 μW @15^{o}C is definitely not sufficient to directly drive the sensor node which requires around 14 mW power. Hence, an energy storage and supply circuit has been implemented to bridge between the source and the load as shown in Fig. 3.8.

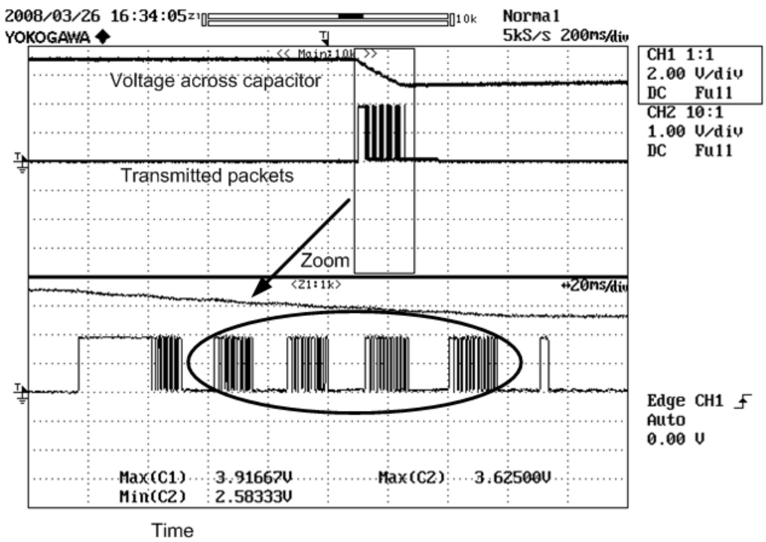


Figure 3.12: Fall detection signal received at base station

Successful transmissions of the fall detected signal can be observed in Fig. 3.12. The charging time of the storage capacitor is very short, simply less than 30 sec intervals with a thermal gradient of approximately 15^{o}C across the harvester. The actual packets transmitted were 5 digital packets over approximately 120 msec, the actual useful energy used equated to 50 ms of active transmission time using 660 μJ and 120 msec of operating time for the other loads using 292 μJ, therefore 952.4 μJ of energy is required to be stored in the capacitor.

3.4 Power Management in Real-time Wireless Sensor Networks

3.4.1 Transmission Power Level Configuration for Wireless Sensor Nodes

In the sensor nodes, the radio consumes a considerable energy compared with the microcontroller and many types of sensors such as temperature, light and humidity. Hence, controlling the energy consumption of the radio is essential. Most of the advanced radio units include both transmitter and receiver, and thus data sending and receiving functions can be performed in one component, a transceiver. In a transceiver, energy consumption of receiving mode is normally fixed at a certain amount depending on the sensitivity of the antenna. The energy consumption of sending mode can be varied by changing the transmission power which is proportional to the communication range. This enables the transceiver to avoid expending higher energy for short range communication.

The low power transceiver ATmel AT86RX230 used in Crossbow IRIS platform provides 15 transmission power levels shown in the Fig. 3.13, which can be programmable.

To evaluate the power consumption of the sensor nodes during the transmitting mode, an experiment is carried out by connecting a 10Ω resistor between a power supply and a IRIS node. The test-bed for the measurement is sketched in Fig. 3.14 .

The communication service provided by TinyOS functions the radio in listen-

Transmission Power Level [TX-Level]	Output Power [dBm]	Output Power [mW]
0	+3.0	2
1	+2.6	1.8
2	+2.1	1.6
3	+1.6	1.4
4	+1.1	1.3
5	+0.5	1.1
6	-0.2	0.955
7	-1.2	0.759
8	-2.2	0.603
9	-3.2	0.479
10	-4.2	0.3802
11	-5.2	0.302
12	-7.2	0.191
13	-9.2	0.1201
14	-12.2	0.06026
15	-17.2	0.01906

Figure 3.13: Transmission power levels of the transceiver ATmel AT86RF230

ing mode as default one whenever the transceiver is turned on, and only change to transmission mode when a send function is called. Therefore, a longest data packet of 28 bytes supported in communication service is transmitted from a node to a base station in order to observe the current drawn by the transmitting node. Fig. 3.15 shows the experimental result which indicates that the reduction of current consumption is corresponding to the decrease of transmission power level and thus, energy consumption is less.

However, the transmission power for energy saving is only significant if a desire number of successful transmission is achieved. Due to interference and channel fading, there might be more data packet lost and the efficiency of the communication is reduced over long distance. The following experiment is conducted

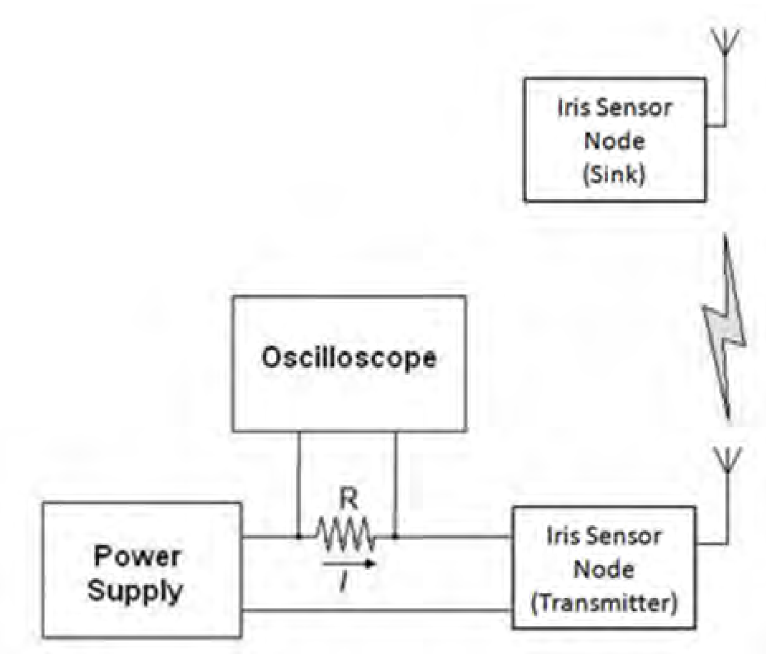


Figure 3.14: Test-bed for the measurement of current consumption at different transmission power levels

to estimate the communicating efficiency with different transmission power level when the transmitting distance is change. An end node is used to periodically send 100 data packets of 28 bytes to a sink connected to a computer. Setting a fixed transmission power level, amount of data packet received is observed at a certain location, and is carried on when the distance to the sink is changed. The observation of communication efficiency is shown in Fig. 3.16, the lowest transmission power level 15 can provide the efficiency of 90% successful transmission within a diameter of about $19m$.

The distance between the transmitting node and the receiving node can be identified by observing the Receive Signal Strength Indicator (RSSI). RSSI value recorded by the transceiver is provided to upper layer for later use through the operating system. The longer distance results in smaller value of RSSI. However, there exists fluctuation in RSSI value because of interference in deployed environ-

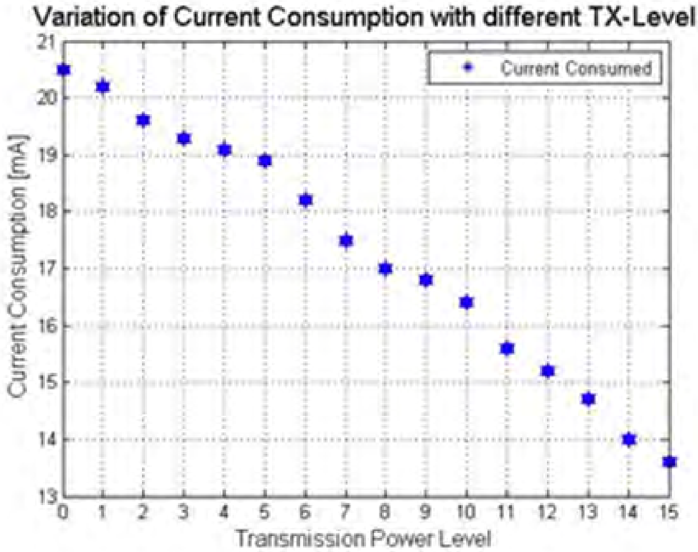


Figure 3.15: Variation of current consumption over transmission power level

ment. Further experiment is needed to figure out the relationship between RSSI and the efficient distance for communication. In this work, an observation of RSSI value over distance which is efficient enough (> 90% successful transmission) to transmitting packet is shown in Table 3.3.

Table 3.3: The relationship between RSSI and efficient distance for communication

Distance (m)	RSSI value (Hex)
0-1.5	$> C$
1.5-12	$5 - 8$
12-161.5	$2 - 5$

This result is used in a linear network which contains nodes located in a line as illustrated in Fig. 3.17. A multi-hopping routing is applied, each node in the network sends its data to the nearest node which is closed to the sink. Initially, every node in the network sends a join message to the sink. Basing on the RSSI value of these messages, the sink identifies the order or the nodes in term of distance to the sink.

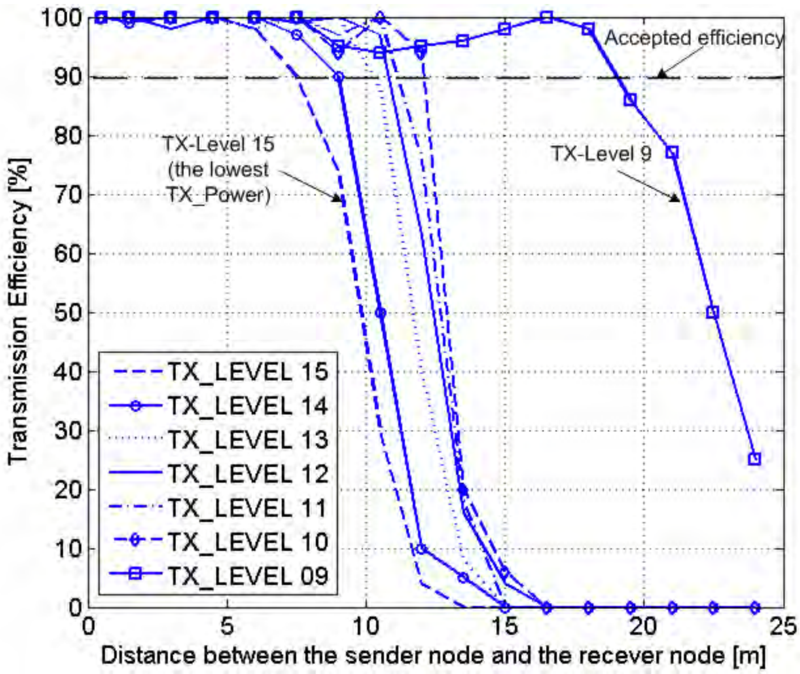


Figure 3.16: Percentage of successful transmission at different transmission power level when changing the distance to receive node

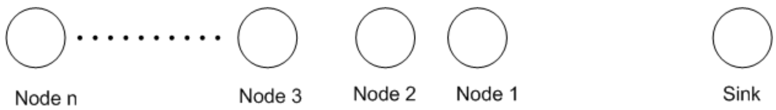


Figure 3.17: N hops linear network

An experiment is conducted with four nodes sending message to a sink. Measurement of power consumption is shown in Table 3.4. 33.7%, 20% and 18% of power consumption is saved when the node only needs to transfer data to its closest neighbour, which is within the range of $0 - 1.5m$, $1.5 - 12m$ and $12 - 16m$ respectively.

3.4.2 Intra-cluster Power Management for Wireless Sensor Networks

In some application scenarios such as fall detection in health condition monitoring systems, indoor monitoring for fire detection or security or machinery condition

Table 3.4: Power consumption of the network with and without adjusting power transmission level

Distance between two nearby node (m)	Power consumption at highest level (default mode) (mW)	Power consumption after adjusting (mW)	Percentage of saving (%)
0-1.5	61.5	40.8	33.7
1.5-12	61.5	49.2	20
12-161.5	61.5	50.4	18

monitoring, the number of sensor nodes in a particular area is known and the nodes are stationary after being deployed, therefore cluster formation can be predetermined. However, the selection of cluster head is still considered carefully to obtain an energy efficient operation of the cluster. This section presents an energy efficient power management mechanism proposed and implemented in IRIS hardware platform for intra cluster data transmission.

a) Assumptions

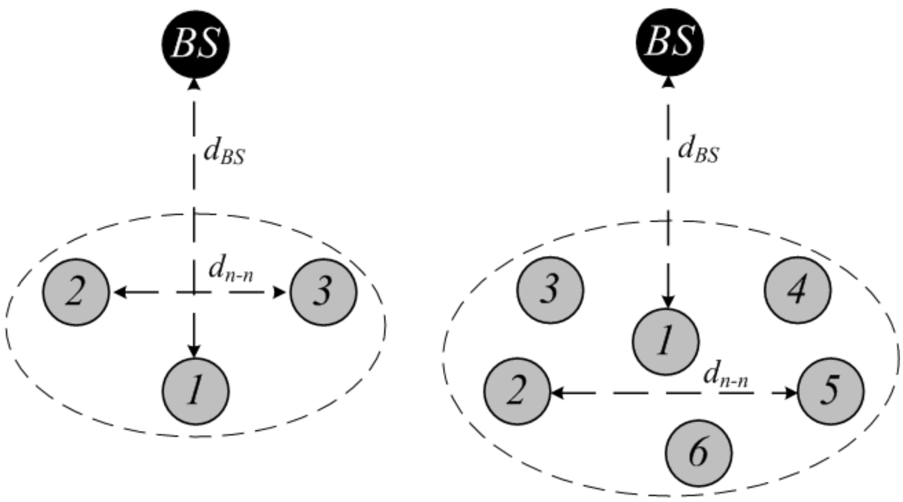


Figure 3.18: Nodes' deployment with the cluster of $N = 3$ and $N = 6$

A scenario in which the sensor nodes are deployed at fixed positions, forming a cluster, to continuously monitor a pre-defined parameter in the environment is considered. The assumptions of network operation are made as follows:

- All the nodes and BS are stationary after being deployed in the field. Every node is capable of operating both in the CH mode or Cluster Member (CM) mode.

- All the nodes measure the environmental parameters at a fixed sampling rate, T_{sample} and send it periodically to the receiver nodes.

- The mean distance between the BS and the cluster, d_{BS} is significantly larger than the distance between any two sensor nodes, d_{n-n} as shown in Fig. 3.18.

- Every sensor node is within a minimum communication range of the others, i.e. each node only needs to use the minimum radio transmission power, P_{tx}^{min} to send data to another node at a success rate of greater than 90 %.

- The BS node is within the maximum communication range of every node, i.e. the furthest node from the BS is possible to send data to the BS by using its maximum radio transmission power, P_{tx}^{max} at the success rate of greater than 90 %.

- Every node is considered to be alive if its battery voltage is above a threshold voltage, V_{th}, representing the residual energy of the node.

b) Power Consumptions of Nodes and Energy Supply

The power of the sensor node is consumed by three main components: *sensor*, *transceiver* and *microcontroller* which carry out three main tasks: sensing, communication, and data processing respectively as mentioned above. While the choice of sensors depends on specific application, a generic hardware platform including a transceiver and a microcontroller can be used for a wide range of applications. In such generic hardware platform, communication task is usually considered to spend most of the node's energy. For example, the MEMSIC IRIS sensor node's radio operation consumes the maximum energy, the radio component's current consumption is $16.5 - 17.5$ mA compared to 8 mA and 8 μA of the microcontroller with the active mode and sleep mode respectively [55].

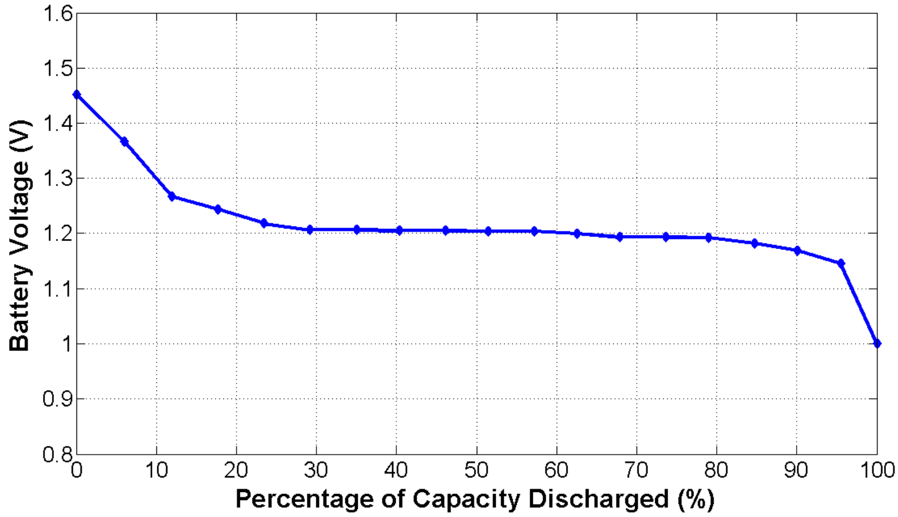


Figure 3.19: The relationship between a fully-charged AA size NiMH battery's voltage and its percentage of capacity discharged, as obtained experimentally

Normally, batteries are used as the energy source powering the sensor node. The IRIS mote can be powered by any combination of batteries with a dc output range of 2.7-3.3V [55]. The energy capacity of this power source can be monitored by the node. Broussely et. al. [109] discuss various battery life prediction methods based on different parameters such as voltage, current and temperature. The most simplistic technique to estimate the battery's remaining charge is by monitoring its voltage while the system is in operation. Typically, a loaded AA battery such as the rechargeable Nickel-Metal Hydride (NiMH) has the voltage against the percentage of capacity discharged as shown in Fig. 3.19.

c) Node and Network Lifetime

With the assumption that a fully charged battery is operating in the range of 100% − 90% of its capacity to power a node j, the voltage change is defined as the difference between its final voltage, V_{Nj}, and its initial voltage, V_{0j} and is given as Equation (3.18). The average rate of voltage change within the interval δt is approximated as Equation (3.19). From V'_j, the lifetime of a node can hence be approximated as the time taken for the node with initial voltage of V_{0j} to meet a threshold voltage, V_{th} as shown in Equation (3.20). The threshold voltage, V_{th} is used to shorten the experiment time. The lifetime of a WSN, L_N is then defined as the duration in which the network is still capable of maintaining a channel of communication with the BS node. L_N can be presented in Equation (3.21) in which (j, BS) indicates that there is a direct communication between node j and the BS.

$$\delta V_j = V_{0j} - V_{Nj} \tag{3.18}$$

$$V'_j = \frac{\delta V_j}{\delta t} \tag{3.19}$$

$$L_j = \frac{V_{0j} - V_{th}}{V'_j} = \frac{V_{0j} - V_{th}}{\delta V_j} \times \delta t \tag{3.20}$$

$$L_N = \max_{(j,BS)} L_j \tag{3.21}$$

d) Network Operations

In this part, the focus is on cluster head selection and power management within one cluster. It is assumed that clusters have been organized in the network, a pre-defined cluster is considered that contains N number of nodes with one node chosen to be the CH. The CM nodes of the cluster transmit their data packets to the CH. The CH compresses the data and sends an aggregated data packet to the BS.

The time in which the network is expected to operate is divided into r number of rounds. Each round consists of a setup phase, a steady-state phase and a rotation phase. At any point in time within the current round, there are one CH node and $N-1$ CM nodes in the cluster. The BS node acts as the destination node at which the information gathered by the network is utilized by the end user.

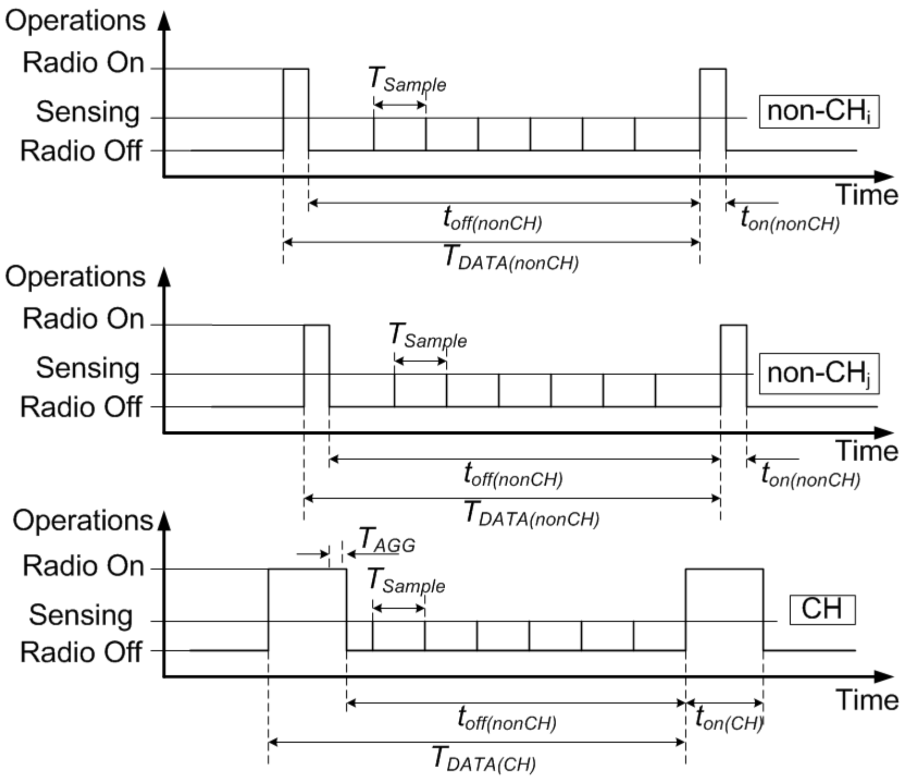


Figure 3.20: The sensing and radio operations intervals for both CH and CM nodes

After being deployed and booted, all nodes enter the setup phase. In this phase, the CH node broadcasts an Announcement (ANNC) Message containing its identity to all other nodes. Meanwhile, all the CM nodes switch on their radio and remain in the listening mode until the ANNC message is received. For the first round by default, node 1 is chosen as the CH node while nodes 2, 3, 4...N are initialized as CM nodes. The setup phase is over once every node successfully identifies the CH node. Node 1 remains as the CH node for one round, which is equivalent to a duration of T_{Round}.

During the steady-state phase, both the CH and CM nodes perform the sensing operation. All CM nodes send their DATA packet to the CH node at a specific interval, T_{DATA}. The DATA packet contains information of the sender, the monitored parameter and the node's battery voltage. The CH receives the DATA packets from all the CM nodes, it processes all the received information, and combines them with its own information and sends an Aggregated data packet (AGG) to the BS at a period of T_{AGG}. It also stores every CM node's most up-to-date battery voltage values in its memory. Based on the composite information contained in AGG, the BS node is able to provide the end user with an overview of the environment in which the WSN is deployed in, instead of just individual data from each node.

$$D_{CH} = \frac{t_{on(CH)}}{T_{DATA(CH)}} \quad (3.22)$$

$$D_{CM} = \frac{t_{on(CM)}}{t_{on(CM)} + t_{off(CM)}} \quad (3.23)$$

where

$$T_{DATA(CH)} = t_{on(CH)} + t_{off(CH)} \quad (3.24)$$

$$T_{DATA(CM)} = t_{on(CM)} + t_{off(CM)} \quad (3.25)$$

In the rotation phase, the CH node selects the next CH based on one of the different CH selection schemes, and inform the chosen node by sending a Baton

(BTN) message to it. Upon receiving the BTN message, this node is initialized as the CH node for the next round, while others are set as CM nodes.

All the nodes in the network then enter the setup phase again and the process is repeated periodically during the network lifetime.

Within each round, all N nodes in the network perform the sensing operation to monitor their surrounding at a common sampling period, T_{Sample} as illustrated in Fig. 3.20. Concurrently, every node also monitors its own battery voltage as an indication of its energy level. Both these parameters are encapsulated in DATA, which is then sent to the CH node.

Since communication task consumes most amount of energy in a node, the radio of every node is switched on only when needed. Additionally, the node's variable radio transmission power is also utilized.

The CM nodes have their radios in the sleep mode for a period of $t_{off(CM)}$ when they perform sensing, and are only switched on periodically with cycle time T_{DATA} as shown in Fig. 3.20. DATA is then sent to the CH node with a minimum radio transmission power, P_{tx}^{min}. Once done, the radio is switched off. All these occur within $t_{on(CM)}$. A CH node's radio, on the other hand, is in the sleep mode for a duration $t_{off(CH)}$. It is then switched on for time $t_{on(CH)}$ to enable it to receive the incoming DATA messages from the CM nodes and send an AGG with a maximum radio transmission power, P_{tx}^{max}, to the BS node. Hence, the CH and CM node's radio duty cycle, D_{CH} and D_{CM}, are given in Equation (3.22) and (3.23), respectively. D_{CH} and D_{CM} are chosen in such a way that D_{CH} is much greater than D_{CM}. The purpose of this selection is to enable the CH node to receive DATA packet from every CM and sending the AGG packet to the BS node successfully.

In the rotation phase, every node ceases all sensing activities and switches on its radio. The CH node of the current round, r, then performs one of the following algorithms to select the next CH node of round $(r+1)$. The CH node identity for the current round is denoted as CH_r and the next CH node is CH_{r+1}. Four CH selection schemes are employed to study the performance of the network.

- ***Fixed-CH Algorithm (FCA)***

FCA defines a fixed node as the CH for the whole network lifetime until this node runs out of energy. This is similar to single hop communication when the CH acts as a relay node which forwards the information from other nodes to the BS.

Algorithm 1 Fixed-CH Algorithm (FCA)

$CH_{i+1} = CH_i$

- ***Sequential Selection Algorithm (SSA)***

This scheme simply lets the CH_r choose the node in a sequential manner based on the node ID to be the CH node in round $(r+1)$.

Algorithm 2 Sequential Selection Algorithm

$CH_{r+1} = CH_r + 1$

- ***Random Selection Algorithm (RSA)***

The CH_i chooses a node indiscriminately with every node j having an equal probability. The CH_{r+1} must be a node that has never performed the CH role in the last $(N - r \bmod N)$ rounds.

Algorithm 3 Random Selection Algorithm (RSA)

```
while CH_{r+1} is not found do
    Generate a random integer number j, 1 ≤ j ≤ N
    if Node j has not been a CH in the last (N − r mod N) rounds then
        CH_{r+1} = j
    end if
end while
```

- ***Voltage-based Selection Algorithm (VbSA)***

Every transmission of DATA from each CM to the CH node contains the updated information of each node's battery voltage. This information assists the CH_i to dynamically choose the node with the highest battery voltage at the end of round i as the next CH node.

Algorithm 4 Voltage-based Selection Algorithm (VbSA)

```
V_rj = battery voltage of node j in round r
Initialize V_max = 0
for j = 1 TO N do
    if node j has not been a CH node in rounds ≤ i then
        if V_rj > V_max then
            V_max = V_rj
            CH_{r+1} = j
        end if
    end if
end for
```

After the identity of the CH_{i+1} has been determined, node CH_i transmits the BTN to node CH_{i+1}. All nodes begin this round $(i + 1)$ by entering the setup phase. This process is repeatedly carried out until $i = N$, after which all N nodes would have performed the CH role once. In the case when an energy harvesting source is applied, the energy of each node can be replenished, and the voltage can be raised up, it is only needed to find the node with the highest battery voltage to become the CH.

Table 3.5: Corresponding scenario and environmental conditions

Scenario	Overall Condition of Environment	Room condition & Meaning
1	$\theta_j < \theta_{th}$ for all j, and $M_j = 0$	0 (Safe)
2	$\theta_j < \theta_{th}$ for all j, and $M_j > 0$	1 (Safe)
3	$\theta_j \geq \theta_{th}$ for any j, and $M_j = 0$	2 (Warning)
4	$\theta_j \geq \theta_{th}$ for any j, and $M_j > 0$	3 (Danger)

e) Application and Experimental Setup

A WSN application that incorporates the hierarchical routing protocol and data aggregation is developed and implemented. This application is to demonstrate a motion and fire detection system in an indoor environment. After deployment, node j continuously monitors the ambient temperature, θ_j by sampling the data obtained from an attached sensor board's thermistor. The threshold temperature above which fire is present is defined as θ_{th}. Besides, one of the nodes is also equipped with a radar, which is utilized as a motion-detection device. This node detects the presence of any motion within the environment given by M_j. Motion is detected within the range of the radar if $M_j > 0$.

The routing protocol and overall operation of this network are discussed in Section 3.4.2 (d). In general, the BS does not need to know which node senses what values of temperature. All it needs to have knowledge of is the big picture of the environment - its average temperature and whether there is a fire and anyone within the vicinity of the environment. All the information are compiled, represented as a room condition message and then attached to the AGG by the CH node. The room condition message is identified together with each environment condition listed in four scenarios in Table 3.5 that are 0, 1 (Safe), 2 (Warning) and 3 (Danger).

Table 3.6: Setting Values for WSN Experiment

Parameter	$N = 6$ nodes	$N = 3$ nodes	Unit
T_{Round}	1800	3600	s
T_{Sample}	400	400	m s
T_{DATA}	5000	5000	ms
T_{AGG}	100	100	ms
$t_{on(CH)}$	1000	1000	s
$t_{on(CM)}$	1.00×10^{-2}	1.00×10^{-2}	s
D_{CH}	2.00×10^{-1}	2.00×10^{-1}	-
D_{CM}	2.00×10^{-3}	2.00×10^{-3}	-
P_{tx}^{max}	-17.2	-7.2	dBm
P_{tx}^{min}	3.0	3.0	dBm
d_{BS}	20	20	m
d_{n-n}	5	3	m
V_{th}	2.700	2.700	V
θ_{th}	323	323	K

From the demonstrative application, the effects of the various CH selection schemes on the network lifetime are evaluated and compared. The network has been experimented using numerous Crossbow IRIS motes, each expanded with an MDA100 sensor board [55] that has a built-in thermistor. The BumbleBee Pulse-Doppler Radar [110] is also attached to one node. Both the application and the hierarchical routing protocol are developed on the TinyOS 2.1.1 operating system [33] with the specified parameter values as summarized in Table 3.6. Each node is powered by two AA NiMH batteries connected via a battery pack.

Before the start of the experiment, the batteries are fully charged to its maximum capacity. The nodes are then positioned within the environment forming the network as shown in Fig. 3.18. The values for the maximum d_{n-n} and d_{BS} are specified in Table 3.6.

The experiment is repeated using $N = 3$ and $N = 6$. For each value of N, the four different CH selection schemes are employed separately. The comparison is made among the VbSA described in Algorithm. 4 and other CH selection schemes that are FCA, SSA and RSA.

e) Experimental Results and Discussions

Fig. 3.21 shows the data received at the base station from the cluster head. The data displayed includes the average ambient temperature measured by all the sensor nodes, the safe condition of the environment, the cluster head ID, the initial and current battery voltage of each sensor node. As shown in Fig. 3.21, the current cluster head is node 1, average temperature is $24^{\circ}C$ and the room condition is 0 that means "Safe" as described in Table 3.5.

In Fig. 3.22, the number of nodes which are alive over the operating time of the network by using different CH selection schemes are compared for a three-node network. Table 3.7 summarizes the time taken for the first node to die, L_j^{first}, and the network lifetime, L_N, by applying the different schemes. It is obvious that the time taken for the first node to die using FCA, that is 3.613 hours, is shorter than the other schemes. The reason is that one node remains as a CH permanently, its energy is severely depleted. Once the CH dies, the communication channel between the network and the BS node is completely lost. Therefore, the network lifetime,

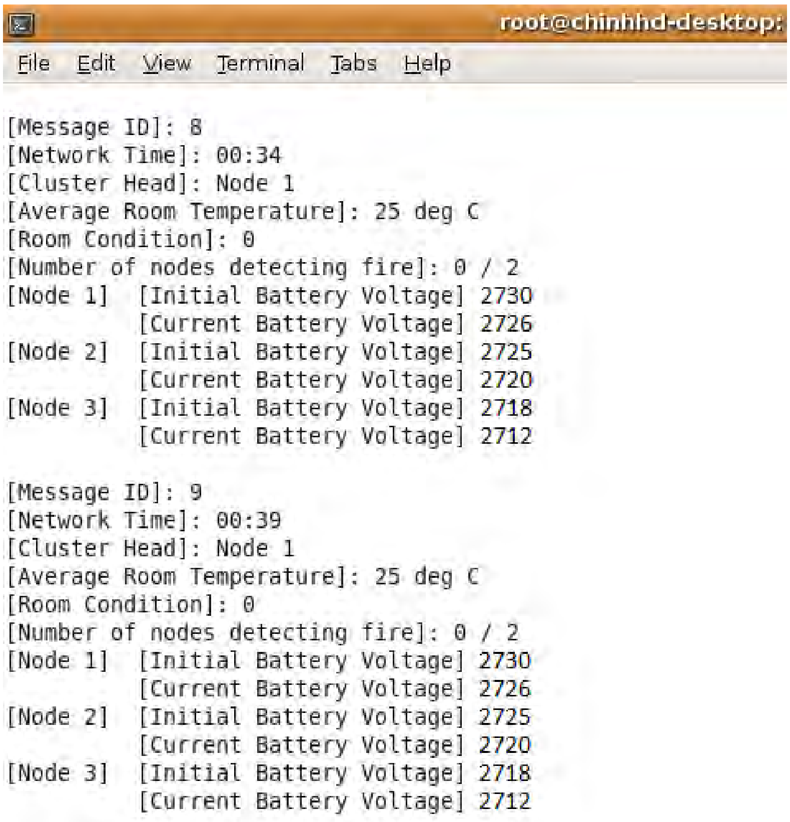


Figure 3.21: The display of data received at the base station

L_N using FCA is equivalent to the time the first node dies, L_j^{first}. The network lifetime when SSA, RSA and VbSA are used is longer than that of network using FCA. These three schemes rotate the cluster head role among all the nodes in the cluster, hence the energy consumption is balanced. It is also observed that VbSA provides the best results in terms of both of the longest time until the first node dies and the most long-lasting network lifetime, 4.462 and 4.811 hours respectively when compared with SSA and RSA. The reason is that VbSA chooses the node with highest battery voltage or the largest residual energy to be the CH, thus it avoids the case when a node having little energy becomes the CH and dies out soon.

By increasing the network size to 6 nodes, the results are illustrated in Fig.

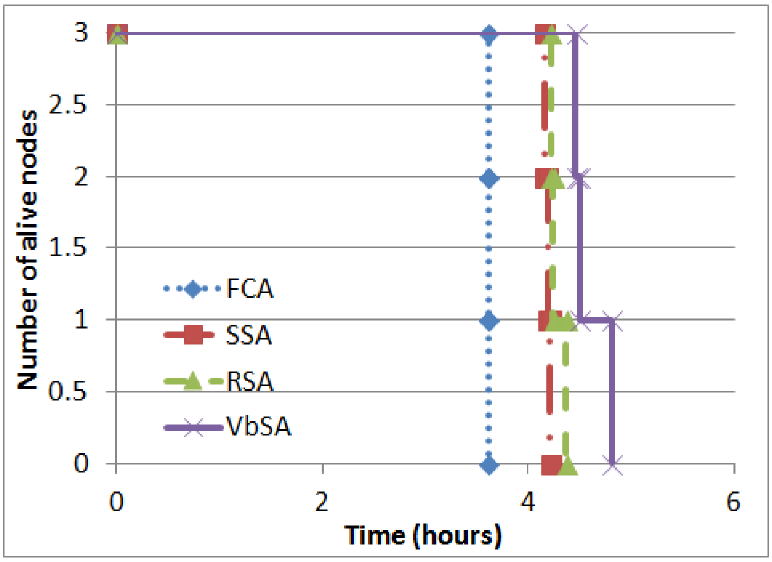


Figure 3.22: Number of alive nodes over time with the cluster of $N = 3$ nodes

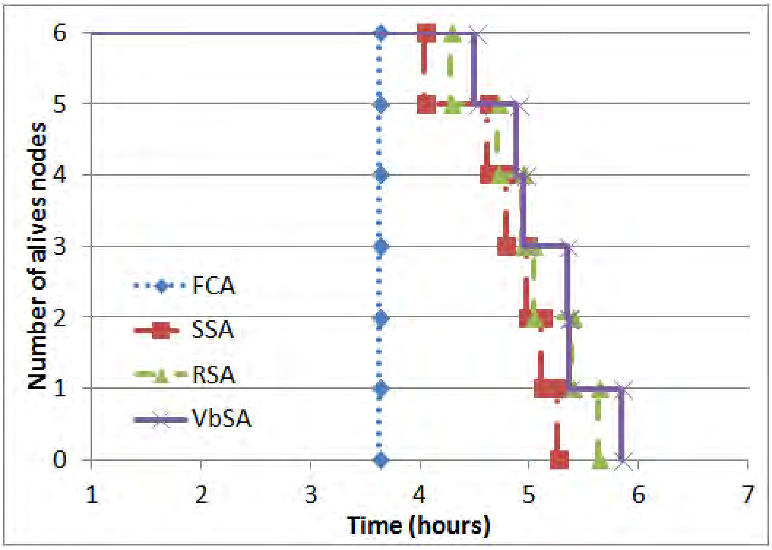


Figure 3.23: Number of alive nodes over time with the cluster of $N = 6$ nodes

3.23, while Table 3.7 summarizes the lifetime for the various schemes. Similarly, the nodes' lifetime as well as the network lifetime of FCA based network are shorter than that of the network using SSA, RSA and VbSA. As compared to a network of three nodes, the RSA shows a better improvement over SSA in terms of the network lifetime, which is 5.633 hours compared with 5.254 hours. This is attributed to the fact that the 6-node network provides more options for the randomized choice of the

Table 3.7: Summary of lifetime for $N = 3$ & $N = 6$

N	CH-Selection Scheme	FCA	SSA	RSA	VbSA
3	L_j^{first} (hours)	3.613	4.155	4.222	4.462
	L_N (hours)	3.613	4.213	4.369	4.811
6	L_j^{first} (hours)	3.621	4.043	4.286	4.500
	L_N (hours)	3.621	5.254	5.633	5.839

next CH node. For both $N = 3$ and $N = 6$, RSA triumphs over SSA because the randomized selection algorithm eliminates the unpredictability in the individual nodes. In all, the best results are obtained when the network employs the VbSA to select a CH with the highest voltage in each round with the network life time of 4.811 and 5.839 hours when $N = 3$ and $N = 6$ respectively. This ensures that the energy-extensive responsibility as a CH node is more efficiently distributed.

3.5 Conclusions

In this chapter, an energy model that is used to evaluate the performance of the network is introduced. A cluster head rotation strategy is simulated to investigate the network lifetime and later is practically implemented for a small scale network consisting of one cluster. Both infinite and finite energy sources, i.e. an energy harvesting source like thermal source and battery, have been studied. A thermal energy source that harvests energy form human warmth has been developed and successfully sustains the sensor nodes in a WBAN. Meanwhile, the information of battery discharge is used to select the CH and evaluate the node lifespan. The power consumption of the sensor nodes for performing different tasks and transferring data at different transmission power level is also investigated and then used to configured the network. The studies have been carried out in this chapter are the initial works of developing energy efficient power management strategies for WSN in the next chapters.

Chapter 4

A Cluster-based Protocol for Wireless Sensor Networks using Fuzzy C-means Protocol

4.1 Introduction

The cluster-based protocols are well-known techniques with special features related to scalability and efficient communication [111]. The concept of cluster-based routing is also utilized to perform energy-efficient routing in WSNs. By grouping the nodes into clusters with the assistance of data aggregation and fusion techniques at the CH, efficient usage of the energy resources is obtained because the overall amount of data to be transmitted to the BS is significantly reduced. The intra-cluster communication enables to reduce the transmission distance of cluster member nodes and therefore reduces energy consumption. Furthermore, duty cycling of the cluster member nodes can be carried out by the CH within the cluster; thus, member nodes are allowed to enter the sleep mode for a longer period of time.

In this chapter, FCM algorithm is adopted to form clusters within the WSNs. The FCM algorithm is first proposed by Bezdek [77] and has been used in cluster analysis, pattern recognition, image processing, etc. This algorithm is a soft partition technique that assigns a degree of belongingness to a cluster for each sensor node. By applying this algorithm, it is expected to overcome the issue of uneven distribution of sensor nodes related with the application of the protocols like LEACH. A uniform creation of clusters in randomly deployed sensor networks is performed. The clusters are groups of sensor nodes which are formed in such a way that the total spatial distance among the sensor nodes within each cluster is minimized. Traffic load is balanced among the CHs within the global network and energy consumption is balanced among sensor nodes in the local cluster. The intra-cluster power management mechanism with time scheduling and battery voltage based CH selection presented in Chapter 3 is incorporated to enhance the energy efficiency of the protocol. Power transmission level is configured accordingly for energy savings within each cluster. The FCM algorithm has been applied in WSNs and proved to be able to enhance the network performance in terms of energy conservation when compared with some conventional methods such as direct communication, Minimum Transmission Energy, LEACH and K-means.

4.2 Preliminaries

4.2.1 Network Assumptions

We consider scenario of application in which sensor nodes are deployed randomly in order to continuously monitor the environment. The information collected by sensor nodes is sent to a base station located far from the deployment field. Furthermore, other assumptions are made as follow:

- Sensor nodes as well as base station are stationary after being deployed in the field.

- The network is considered homogeneous and all of the sensor nodes have the same initial energy.

- Each sensor node knows its own geographical position.

- All nodes measure the environmental parameters at a fixed rate and send it periodically to the receiver nodes.

- The radio channel is symmetric such that energy consumption of data transmission from node A to node B is the same as that of transmission from node B to node A.

Each sensor nodes can operate either in sensing mode to monitor the environment parameters and transmit to the base station or cluster head mode to gather data, compress it and forward to the BS.

4.2.2 Energy Consumption of Cluster-based WSNs

As mentioning in Chapter 3, in most cases of application, among its components consuming power such as data processing unit, sensing unit, memory storage and transceiver, the radio communication unit takes the most energy consumption.

Therefore, we mainly consider the energy consumption of the sensor nodes for communication within the network.

Both the free space and multipath fading channel models as described in Equation (3.1), (3.2) and (3.3) in Chapter 3 are used to compute energy dissipated during the process of transmitting and receiving information, E_{Tx} and E_{Rx} respectively. Another parameter is also taken into account is the data aggregation energy consumption denoted as E_{da}.

Considering a N node network partitioned into c clusters, the average number of nodes in a cluster is N/c, the energy consumption of the CH to receive message from the CM is

$$E_{Rx-CH} = lE_{elec}\left(\frac{N}{c} - 1\right) \tag{4.1}$$

and to aggregate data into a l-bit message and send it to the BS is

$$E_{Tx-CH} = lE_{da}\frac{N}{c} + lE_{elec} + lE_{mp}d_{toBS}^4 \tag{4.2}$$

where d_{toBS} is the average distance from one CH to the BS. Assuming that the distance from the non cluster head node and the CH is short, the energy consumed by the CM node to transmit a l-bit message is

$$E_{Tx-nonCH} = lE_{elec} + lE_{fs}d_{toCH}^2 \tag{4.3}$$

where $d_{toCH} \approx \frac{1}{2\pi}\frac{M^2}{c}$ [47] which is the average distance from one node to its CH and M is the network diameter. Thus, the total energy dissipated in the network during a round of collecting data and transmitting to BS is

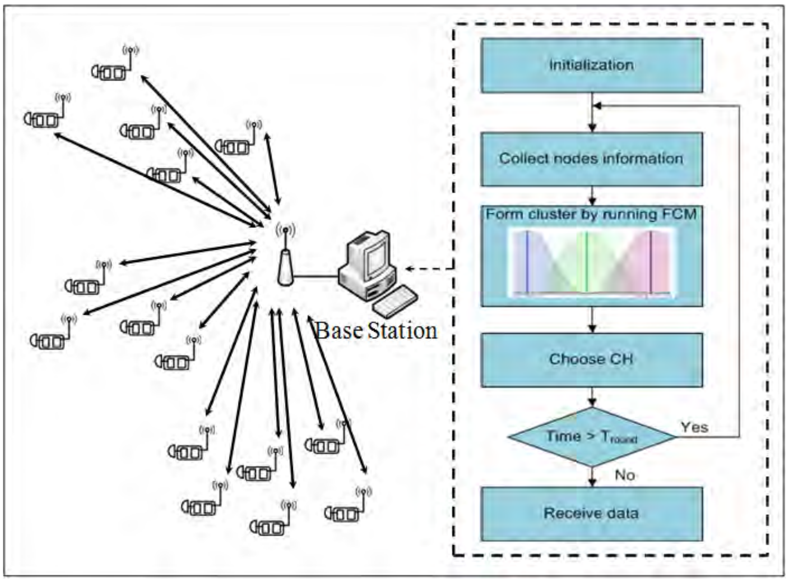
$$\begin{aligned} E_{round} &= l\,(2N\,E_{elec} + N\,E_{da} + cE_{mp}d_{toBS}^{4} \\ &\quad + N\,E_{fs}d_{toCH}^{2}) \\ &= l\,(2N\,E_{elec} + N\,E_{DA} + cE_{mp}d_{toBS}^{2} \\ &\quad + N\,E_{fs}\frac{1}{2\pi}\frac{M^{2}}{c}) \end{aligned} \tag{4.4}$$

This total energy consumption consists of the average energy dissipated by data transmission of the CM nodes and CHs and the energy consumption for data collection and fusion of the CHs.

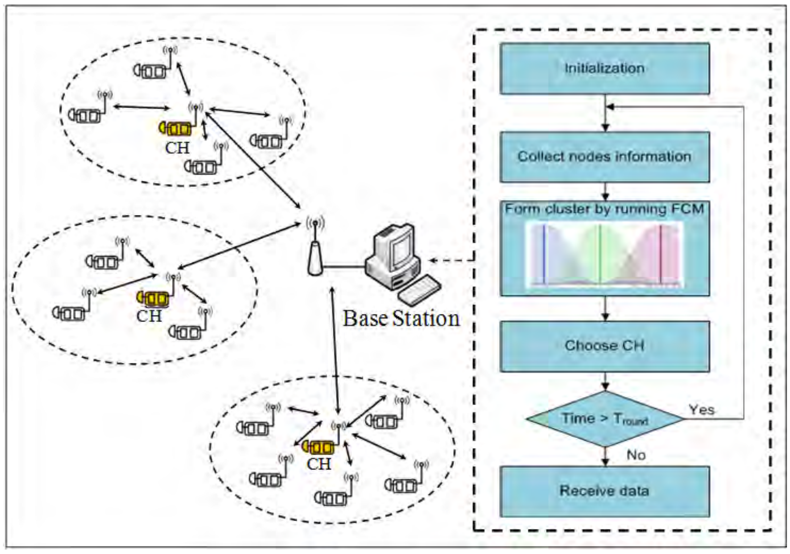
4.3 Fuzzy C-Means Algorithm

The proposed clustering protocol includes two phases: *setup phase* and *data transmission phase* as shown in Fig. 4.1. The *setup phase* performs two tasks: cluster formation and cluster head selection. Meanwhile, during the *data transmission phase*, sensor nodes within each cluster transfer data towards the BS via the selected CHs; the CHs perform data aggregation and fusion, then they send the compressed data to the BS.

In this chapter, the FCMCP using the FCM algorithm is adopted to form the cluster. The FCMCP is a centralized clustering mechanism, the BS computes and allocates sensor nodes to clusters according to the information of their geographical location and the CH is assigned to the node having the largest residual energy within the cluster. One method of evaluating the residual energy of each node is



(a) *Setup phase*: sensor nodes and the BS exchange information for network control.



(b) *Data transmission phase*: data packets are transferred from sensor nodes to the CH and from CH to the BS.

Figure 4.1: The operation of the WSN using FCM.

to measure its battery voltage that is described further in Section 4.6. A network of N sensor nodes which is partitioned into c clusters: $C_1, C_2, ..., C_c$ is considered. The purpose of the cluster formation in this protocol is to minimize the following objective function:

$$J_m = \sum_{i=1}^{c} \sum_{j=1}^{N} u_{ij}^m d_{ij}^2 \tag{4.5}$$

where

u_{ij} is node j's degree of belonging to cluster i,

d_{ij} is the distance between node j and the center point of cluster i

and m is the fuzzyfied parameter.

The degree u_{ij} of node j with respect to cluster i is calculated and fuzzified with the real number $m > 1$ as below:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} (\frac{d_{ij}}{d_{kj}})^{\frac{2}{m-1}}} \tag{4.6}$$

The distance between the sensor node and the center point is the Euclidean distance. By achieving minimization of the spatial distance, the distribution of the nodes into clusters is optimized and thus energy balance can be achieved.

Phase 1a: Cluster formation

In an application scenario, N sensor nodes are deployed randomly in the field with an area of $M \times M\ m^2$ is being considered. After being deployed, these sensor nodes send an advertisement message to the BS with the information of their geographical location and residual energy; based on the information the BS calculates the cluster centers and allocates sensor nodes to the cluster using FCM algorithm.

Each node is assigned a degree of belongingness to a cluster rather than completely being a member of just one cluster. Therefore, the nodes closer to the boundary of a cluster may become a member of the cluster with a degree approximating the degree of belongingness to the neighboring cluster. The formation of the clusters is carried out by following the Algorithm. 5.

Algorithm 5 FCM algorithm for cluster formation

for $j = 1$ to N **do**
  Node j is given the coefficient u_{ij} for being a member of cluster i
end for
repeat
  for $i = 1$ to c **do**
    Compute the centroid of each cluster

$$pos(center_i) = \frac{\sum_{j=1}^{n} u_j^m pos(node\ j)}{\sum_{j=1}^{c} u_j^m}$$

  end for
until the algorithm is converged

The convergence is achieved when the difference between the fitness values in two consecutive iterations is less than a threshold or a sufficient number of iterations are reached.

Phase 1b: Cluster head selection

After forming the clusters, the BS chooses the node with highest residual energy within each cluster to be the CH. Then an assignment message is being sent to every sensor node in the network containing the information of the cluster it belongs to as well as the time schedule to transfer data. Once the assignment message reaches a sensor node, the node dispatches this message and stores the identification of the CH in its memory to forward data during the *data transmission phase*. The process of cluster formation and cluster head selection is repeated periodically during the network operation.

Phase 2: Data transmission

Once all the nodes receive the assignment message, and the transmission schedule is initialized, the sensor nodes start to perform the sensing task and transmit data to the CHs. Transmission power level of cluster member nodes is optimized because of the minimum spatial distance to the CHs achieved by FCM algorithm. Furthermore, by using time scheduling for data transmission, cluster member nodes only need to turn on their respective radio components during the transmission, and turn off after finish transmitting the data. Data aggregation and fusion are carried out at the CHs, and only the compressed data packet is sent to the BS. Therefore, the amount of information transmission is reduced and thus results in the reduction of energy consumption.

4.4 Simulation of FCM cluster-based WSNs

4.4.1 Experiment 1 - Network lifetime assessment with different protocols

In this experiment, FCMCP is evaluated by simulating a 300 node network with MATLAB. The comparison of Direct Communication, MTE[112], LEACH, K-means clustering and FCMCP are also presented.

Sensor nodes are deployed in an area of $250 \times 250 m^2$ shown in Fig. 4.2, the BS is located at (125 m,-75 m). The number of clusters, which is calculated by using the

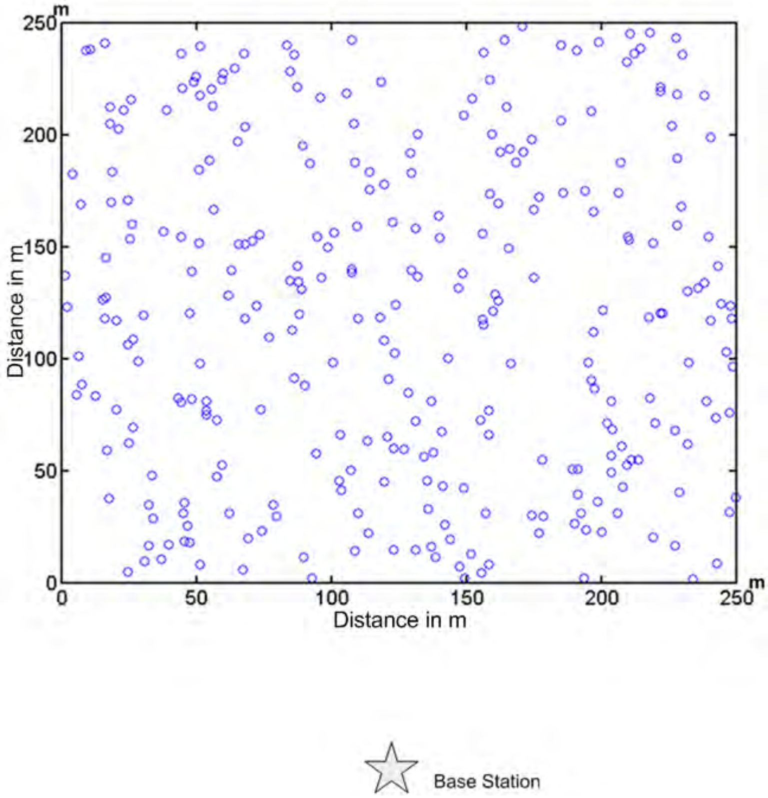


Figure 4.2: Deployment of sensor nodes into monitored field

Equation (4.7)[47], is 5 with the average $d_{toBS} = 200m$. Each sensor node transmits a 4000 bit message with 150 bit header to the cluster head at each round. The initial energy supplied to each sensor nodes is $2\,J$. During the computational experiment, the following values of some above parameters in the energy model given in Chapter 2 are used: $E_{da} = 5nJ/bit/message$, $E_{elec} = 50nJ/bit$, $E_{fs} = 10pJ/bit/m^2$, and $E_{mp} = 0.0013pJ/bit/m4$ [47].

$$c_{opt} = \frac{\sqrt{N}}{\sqrt{2 * \pi}} \sqrt{\frac{E_{fs}}{E_{mp}} \frac{M}{d_{toBS}^2}} \tag{4.7}$$

Fig. 4.3 and Fig. 4.4 show the cluster structure of the network while using LEACH and FCMCP respectively. It is found that the distribution of clusters in LEACH is not uniform, some clusters consist of a huge number of nodes spread out in a large area whereas the others have a few as shown in Fig. 4.3.

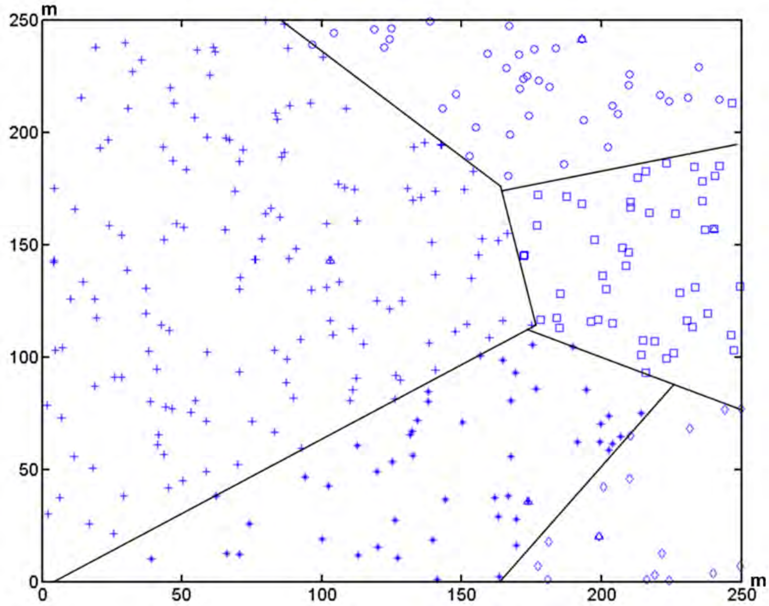


Figure 4.3: Cluster formation with LEACH protocol at a arbitrary round

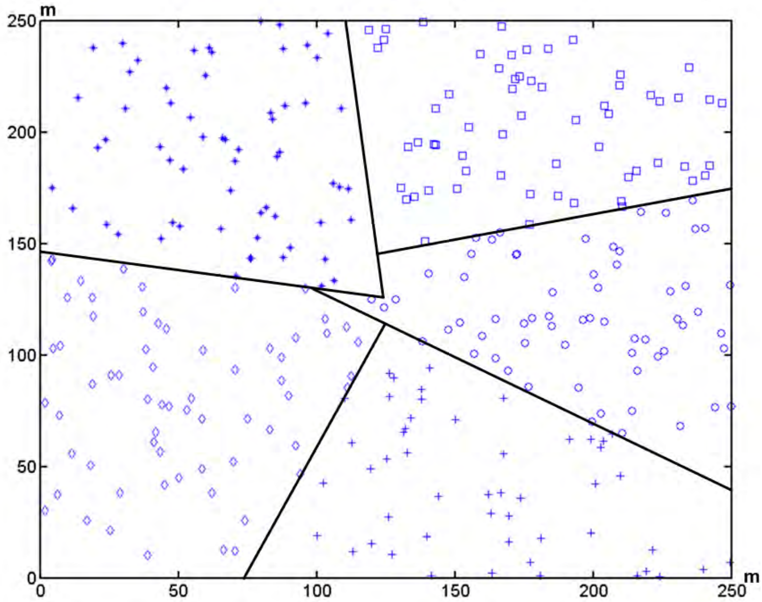


Figure 4.4: Cluster formation with FCMCP

Hence, in some rounds of operation in LEACH, the CH of the group with huge number of nodes has to suffer from heavy traffic load which causes significant power consumption of the CH and high chance of wasting energy due to data collision.

Moreover, many sensor nodes do not have the CH in their proximity, so more

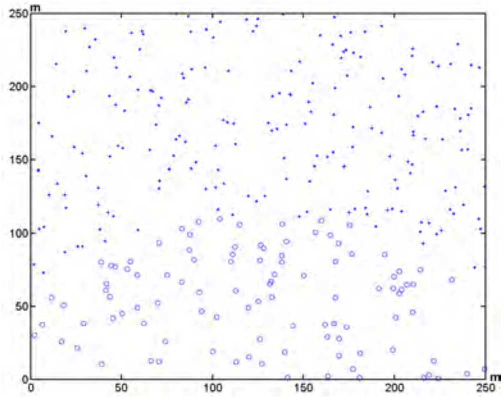


Figure 4.5: Distribution of dead nodes (dots) with Direct Communication after 300 rounds

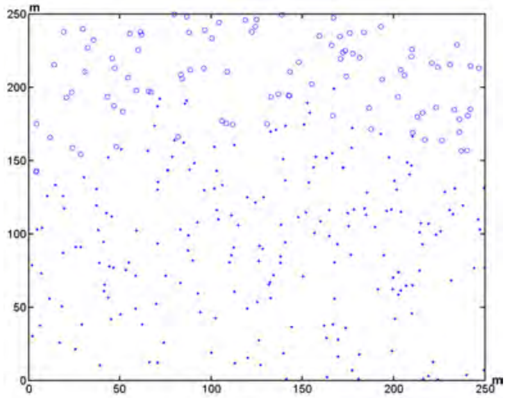


Figure 4.6: Distribution of dead nodes (dots) with MTE after 300 rounds

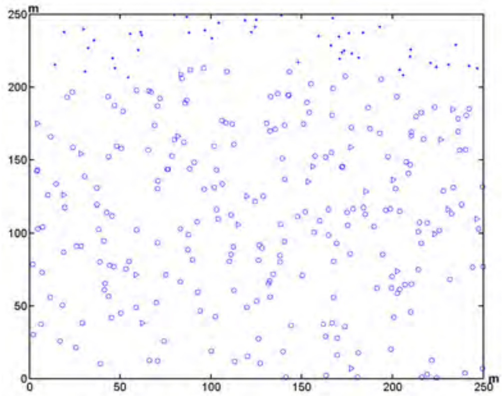


Figure 4.7: Distribution of dead nodes (dots) with LEACH after 700 rounds

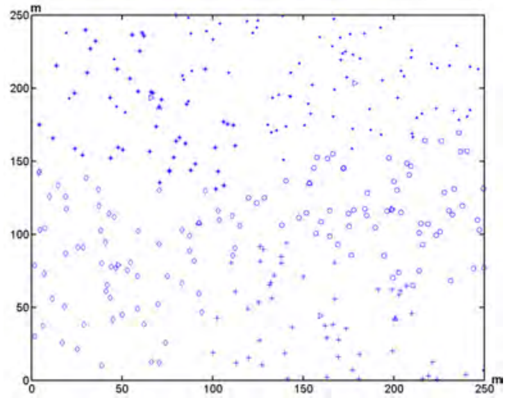


Figure 4.8: Distribution of dead nodes (dots) with FCM after 1600 rounds

energy is required to transmit data from these nodes to CHs over longer distance. Meanwhile by using FCM the clusters has better formation where mean distance from each node to the cluster is minimized. It is more efficient to balance the load of network and to distribute the nodes among clusters by using FCM.

The distribution of alive and dead nodes in different cases is presented in Fig. 4.5, Fig. 4.6, Fig. 4.7 and Fig. 4.8. In the case of Direct Communication, the sensor nodes send data to the BS directly, it is seen that the nodes, further away from the BS, consume more energy to transmit data and thus have a shorter life than that of nodes closer to the BS. Meanwhile, MTE protocol routes messages

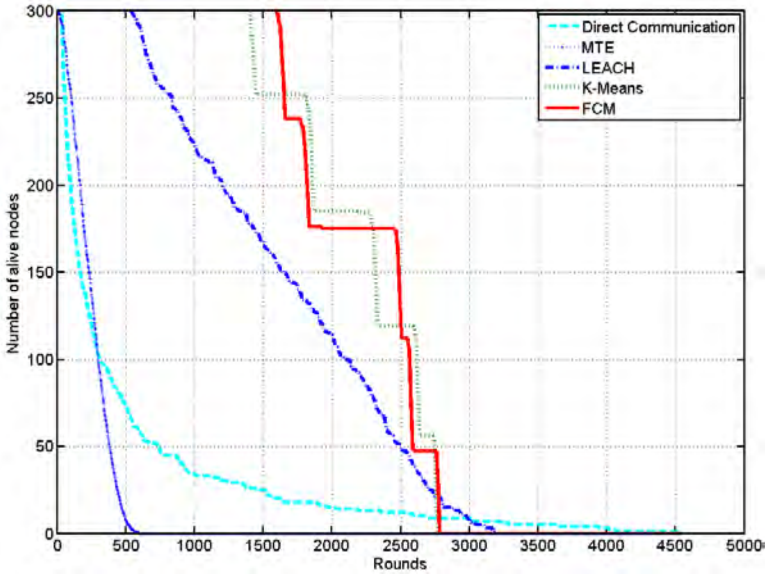


Figure 4.9: Number of node alive over the time with different cluster based protocols

from a node through its nearest neighbor toward the BS, in this case, nodes nearer to the BS have to relay a huge amount of data, thus they are run out of energy earlier than the further nodes

In LEACH, although the nodes further away from the BS have higher chance to run out of energy first, the energy balance among the nodes assists LEACH network to obtain a larger covered area after a longer time of operation, 700 rounds compared with 300 rounds of operation in the case of using Direct Communication and MTE. The lifetime of the network when FCM is employed is upto 1600 rounds. This is much higher than the other protocols, thus it guarantees the sensing of a larger area of the field by sensor nodes.

Table 4.1: Duration of time up to the first node dies in the network

Protocols	Direct	MTE	LEACH	K-Means	FCM
Lifetime of the first dead node (rounds)	11	33	541	1396	1598

In Fig. 4.9, the number of alive nodes over the operating time of the network by using different protocols is compared. In our work, the performance of Direct Communication, MTE, LEACH, K-means and FCMCP are studied. The different duration of time up to the first dead node by applying different protocols is given in Table 4.1. It can be seen that the lifetime of the network with 100% nodes alive when using FCMCP is much longer than the network lifetime when LEACH is employed, and also the lifetime of network with K-Means protocol.

4.4.2 Experiment 2 - Energy consumption evaluation within the deployment network

We studied the average energy dissipated within the network in the second experiment. 100 sensor nodes are randomly deployed in areas with different diameters with 1 J initial energy of each node. The simulation is run to observe the changing of energy consumption with the variation of network diameter and different values of electronic energy. Fig. 4.10 shows the average energy consumption of network with different protocols over the diameter of the network after 200 rounds. With the small network diameter, energy consumption of the network by using different protocols are almost the same. However, when the network diameter increases, FCM outperforms MTE, Direct Communication, LEACH and K-Means. This is because of the better cluster distribution and traffic load balance among the network achieved by using FCMCP as mentioned above.

Another energy consumption comparison made with the same size of network among different protocols is shown in Fig. 4.11, Fig. 4.12 and Fig. 4.13. Once the electronics energy or the network diameter increases, FCM network consumes less

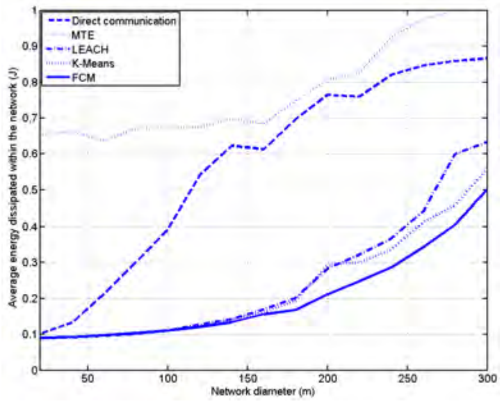


Figure 4.10: Average energy dissipated within the network over the network diameter after 200 rounds

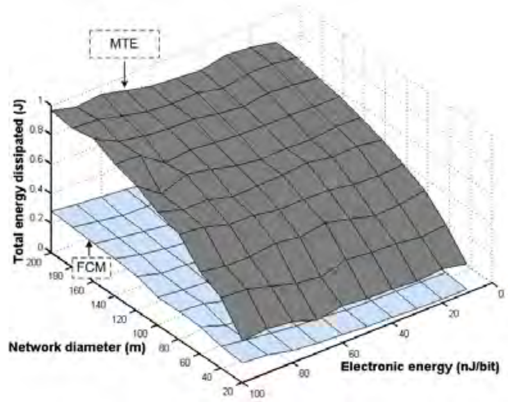


Figure 4.11: Average energy dissipated within the network by using MTE and FCMCP over the network diameter and electronics energy after 200 rounds

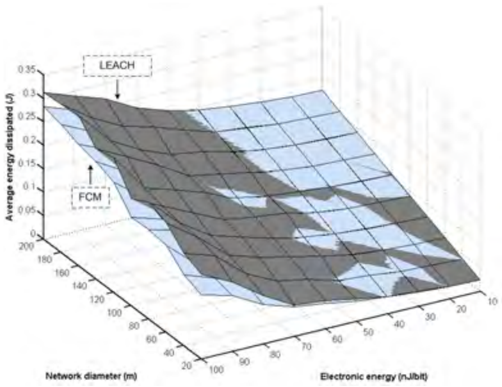


Figure 4.12: Average energy dissipated within the network by using LEACH and FCMCP over the network diameter and electronics energy after 200 rounds

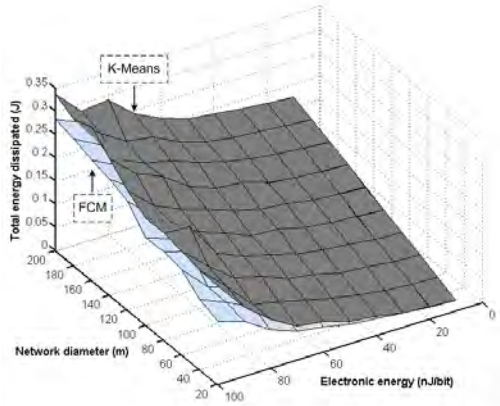


Figure 4.13: Average energy dissipated within the network by using K-Means and FCMCP over the network diameter and electronics energy after 200 rounds

energy than the network with MTE in Fig. 4.11, with LEACH in Fig. 4.12 and with K-Means in Fig. 4.13. The improvement of FCMCP is much clearer when the deployment field is larger and the electronic energy grows.

4.5 Design and Implementation of the protocol in a hardware platform

The FCMCP is realized and implemented in real-life application using Crossbow's IRIS hardware platform [55] supported by the TinyOS operating system. A scale-down network consisting of a smaller number of IRIS nodes is used as test-bed to evaluate the protocol. An IRIS sensor node consists of a low power microcontroller, Atmel ATmega1281, and a transceiver, AT86RF230 with an extension connector to interface with sensor boards.

Fig. 4.14 illustrates the framework for designing the network protocol. The BS and sensor nodes exchange the network configuration information and store in their memory for subsequent usage. At the sensor nodes level, there are two main components: **Data Transmission Engine** and **Cluster Setup Engine**. The **Cluster Setup Engine** contains three modules: *Cluster Information Update*, *Data Transmission Schedule* and *Node Information Update*. The present status of the nodes such as location and battery voltage are obtained by *Node Information Update* module, and then provided to the BS where it is stored in a node information table. The *Data Transmission Schedule* module creates a schedule for the node to transfer data to the CH based on the setting values in the received ASG message. The *Cluster Information Update* module updates the information of the cluster configuration such as the ID of CH and the number of cluster members. The **Data Transmission Engine** takes the responsibility of performing data aggregation and transmitting data packets with assistance of the *Data Aggregation* and *Packet Transmitting* modules respectively. At the BS level, there are also two main components: **Cluster Management** and **Data Acquisition**. The HSA is integrated in the *Clustering Algorithms* module of the **Cluster Management** component. This module takes the responsibility of selecting the CH and forming the clusters, the calculation results are kept in the *Cluster Information Table* module as shown in Fig. 4.14. The **Data Acquisition** component at the BS is used for processing and storing data received from the sensor nodes.

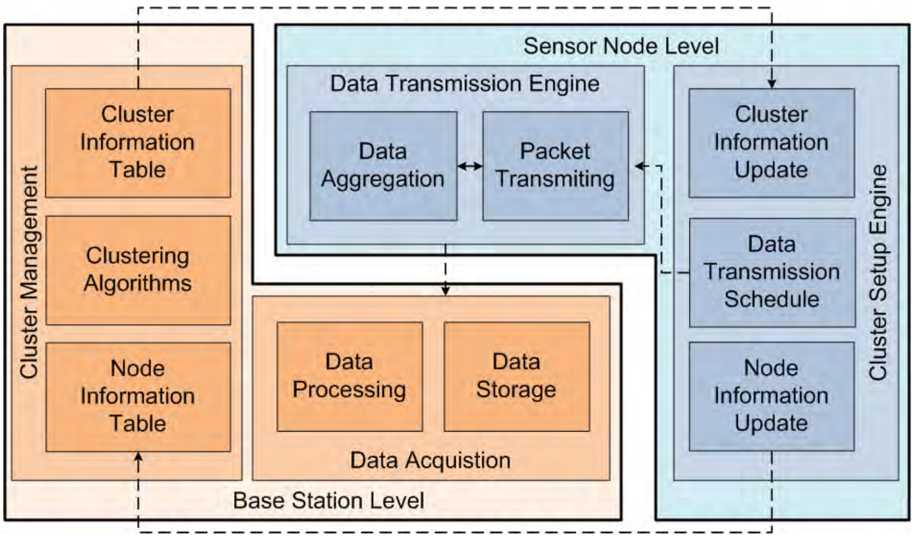


Figure 4.14: Diagram of the routing frame work for a centralized cluster-based protocol.

The FCM algorithm as described in Section 4.3 is used in the clustering algorithm module. The BS layer of the protocol is written in Java and run on a Linux based computer. Information of the sensor nodes sent to a BS node is forwarded to the computer via a serial port.

The sensor node layer of the protocol is developed with the support of the TinyOS 2.1.1 operating system. Fig. 4.15 illustrates the main components of the routing layer for the proposed cluster-based protocol implemented on the hardware platform. This routing layer is built on top of the TinyOS **ActiveMessageC** which uses the Carrier Sensing Multiple Access-with Collision Avoidance (CSMA/CA) mechanism for avoiding packet collision at the receiver during the *data transmission phase*. This is the main difference from many existing cluster-based protocols

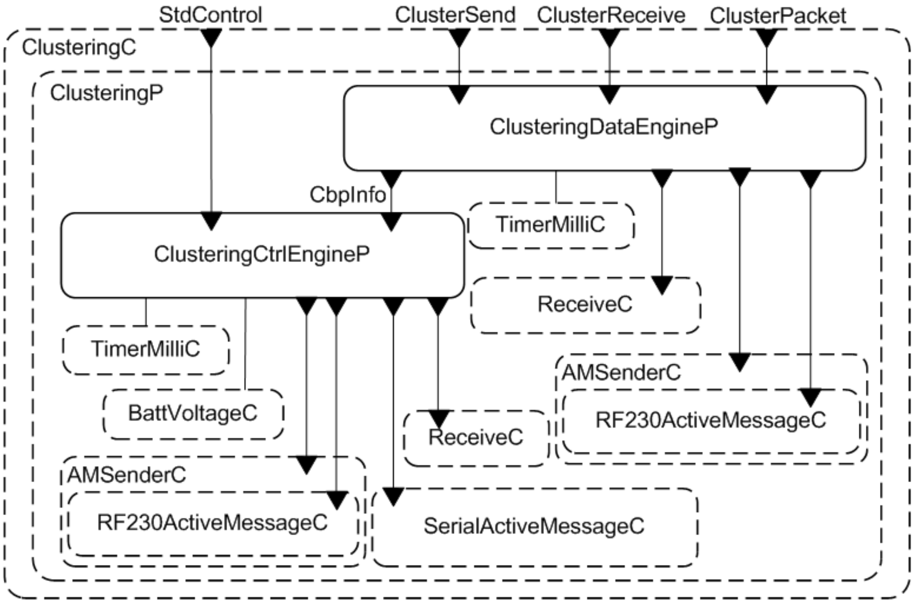
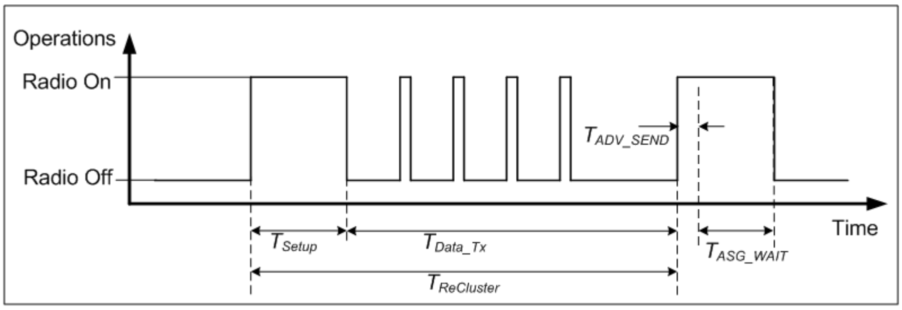
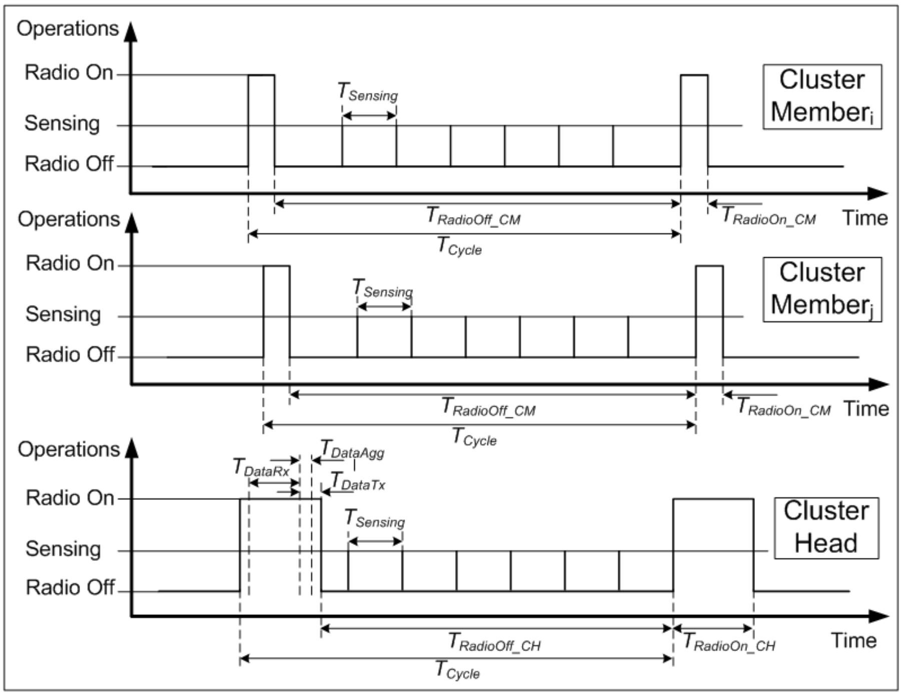


Figure 4.15: Main components of the routing layer.

that use TDMA mechanism. Although it might be more efficient in terms of energy consumption, TDMA requires strict time synchronization among the nodes within a cluster and not suitable for applying in the networks where the topology changes changes frequently [113]. Meanwhile, CSMA/CA is widely applied and integrated as a built-in function in most of the transceivers, especially for WSNs, since it is simple, flexible, and robust [114]. By using this mechanism, a loose synchronization scheme within a cluster can be carried out that is much easier and possible for practical implementation. The **CluteringCtrlEngineP** component plays the role of sending the advertisement message and receiving the assignment message during the *setup phase* of cluster-based network. During *data transmission phase*, it provides information of the CH as well as the data transmission time schedule to the **ClusteringDataEngineP** component. The component **ClusteringDataEngineP** contains interfaces to send and receive data that are used by the application layer. Data aggregation can be implemented at the application layer since the type of data is dependent on specific application.



(a) Time schedule for re-organization of the network.



(b) Time schedule for data transmission within each cluster.

Figure 4.16: The time schedule of the network operation

The operation of the sensor nodes in the *setup phase* is illustrated in Fig. 4.16a. The *setup phase* happens during the period of time T_{Setup}. At the beginning of this phase, each sensor node sends an advertisement message to the BS node connected to a computer within the duration T_{ADV_SEND}. The structure of this message is shown in Fig. 4.17, which includes the location and battery voltage of the sensor nodes. Once the advertisement message is transmitted successfully, the sensor node keeps its radio on to wait for the information of the network organization sent from the BS. If this information is received within the time duration T_{ASG_WAIT}, the *data transmission phase* is started. Otherwise, an error is considered to occur, and the sensor node sends another advertisement message to request for joining the network. The *setup phase* is repeated periodically at the sensor nodes after the time duration $T_{ReCluster}$.

```
typedef nx_struct advMessage{
  nx_am_addr_t nodeID; // node ID
  nx_uint16_t posX;    // x coordinate
  nx_uint16_t posY;    // y coordinate
  nx_uint16_t battVoltage; // battery voltage
} Node_advMsg_t;
```

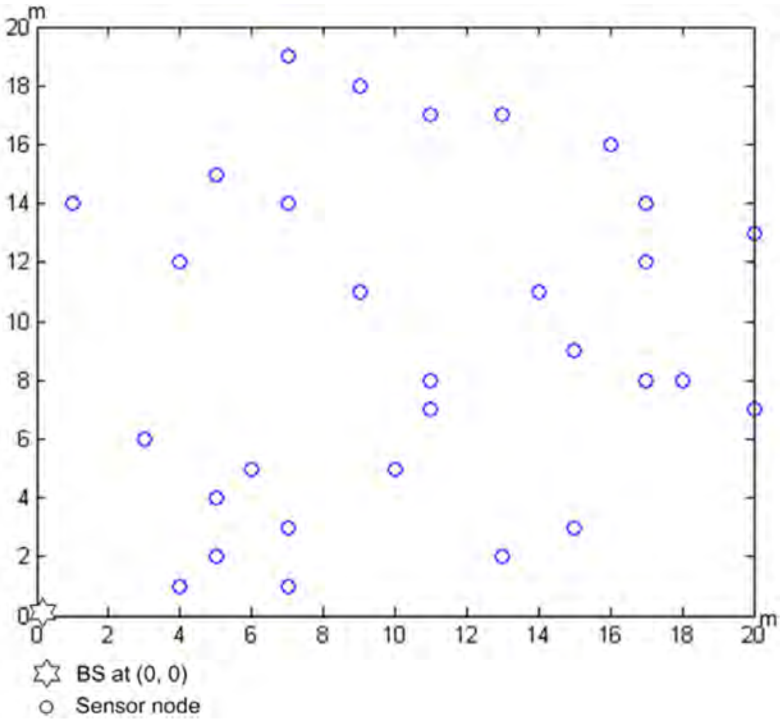
Figure 4.17: The structure of an advertisement message

The sensor node information received at the BS node is forwarded to the computer via a serial port. Once the advertisement messages from all the nodes are collected, the FCM algorithm is executed to calculate the cluster formation and then the CH for each cluster is selected based on the residual energy which is estimated by using the battery voltage of the sensor nodes. When the computation is completed, the BS node sends assignment messages to each node in the network which contains the information of the CH as well as the time schedule for data transmission and cluster re-organization. The structure of the assignment message is shown in Fig. 4.18.

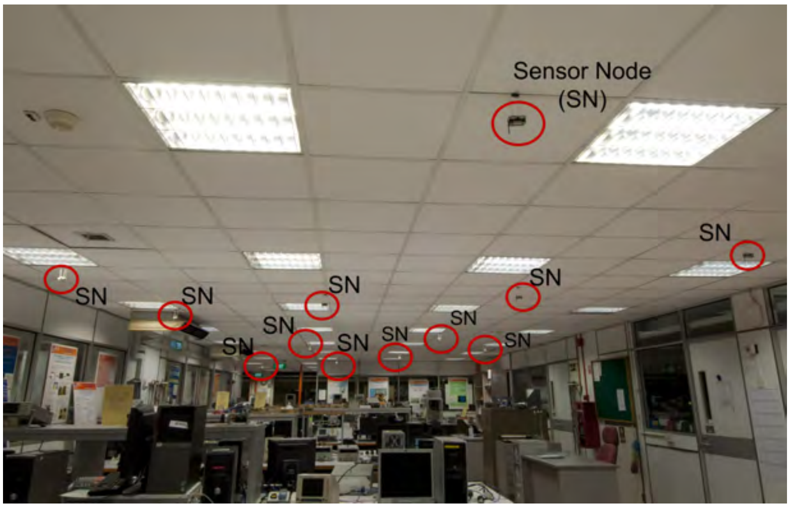
```
typedef nx_struct asgMessage{
  nx_am_addr_t CHID; // ID of the cluster head
  nx_uint8_t NoMember; // number of member within the cluster
  nx_uint16_t txOnDataInterval; // maximum time to transmit one data packet
  nx_uint16_t txTotalDataInterval;  // interval between two data packets
  nx_uint16_t reClusterFormationInterval;  // time to reorganize the network
  nx_uint8_t txPower; // data packet transmission power
} BS_asgMsg_t;
```

Figure 4.18: The structure of an assignment message

After the sensor node receives its assignment message, it starts *data transmission phase* based on the role assigned, i.e., as a cluster member or as a CH. Fig. 4.16b illustrates the time schedule for data transmission at the CH and cluster members established by using the information in the assignment message. During this phase, the cluster member performs the sensing task at a sampling time $T_{Sensing}$. The sensing task is carried out within the time duration $T_{RadioOff\_CM}$ when the radio component of the cluster member is off. When the sensing task is completed, the cluster member pre-processes and combines the sensing measurements if required, then sends its data to the CH within the time $T_{RadioOn\_CM}$. Meanwhile, the CH gathers data from all its member nodes, aggregates the information, and then transmits the compressed data packet to the BS. The data reception is complete when the CH gets data packets from all its members or the maximum time to receive T_{DataRx} is reached. As shown in Fig. 4.16b, the data transmission from the member node is performed within the period of time that the CH turns on its radio component for receiving data. After completing its data transmission, the cluster member can turn off its radio to save energy. Meanwhile, the CH needs to keep its radio on so that data packets from all the alive members of the cluster are received or the maximum time interval given to receive data, T_{DataRx} is reached. The radio component of the CH can be turned off once it finishes processing the received data and sends the compressed packet to the BS with the maximum duration for data aggregation and data transmission is $T_{DataAgg}$ and T_{DataTx} respectively.



(a) Overall diagram of the network deployment.



(b) Experimental setup for the research work.

Figure 4.19: Network deployment in the environment.

4.6 Network and Sensor node configuration

The FCMCP proposed above can be applied for environment monitoring. In this work, it is employed to assist the WSN that monitors the temperature of an indoor environment in order to detect a fire event. The sensor nodes are deployed in a room of $20m \times 20m$ area as shown in Fig. 4.19a and Fig. 4.19b. Each node is equipped with the sensor board MDA100CB with the thermistor YSI 44006 [115]. The sensor nodes perceive temperature of the environment and send the measured temperature values to the BS via the CHs. Since the objective is to detect the fire event, the BS does not need to receive sensing values from all the sensor nodes. The temperature value in the data packet from each sensor node in a cluster is retrieved and processed at the CH level. In order to reduce the computational burden for the CH, the raw value read by the microcontroller is converted into Celsius degree before sending, the conversion is performed by using the formula given in [115]. Only the maximum, minimum and average temperature values of the cluster is included in the data aggregation packet sent from the CH to the BS as shown in Fig. 4.20.

```
typedef struct CHAggregatedMessage{
  nx_uint16_t aveTemperature;
  nx_uint16_t maxTemperature;
  nx_uint16_t minTemperature;
} aggregation_message_t;
```

Figure 4.20: The structure of data aggregation packet sent by the CH

The IRIS sensor node can be powered by any combination of batteries with a dc output range of 2.7-3.3V. The most simplistic technique to estimate the batteries' state-of-charge is to monitor its terminal voltage while the batter is delivering its load as presented in Chapter 3. Before starting the experiment, the batteries are fully charged to their maximum capacity with the respective voltage, V^{init}. Every cycle of re-organizing the network, each sensor node measures its present battery voltage, V^{pre} and attaches this voltage information to the advertisement message.

In the first cycle of the network operation, the present battery voltage is considered as the initial voltage of the respective node: $V^{init} = V^{pre}(1)$. Both the values of V^{init} and V^{pre} are kept in a memory table of the BS layer programme. A sensor node is identified to run out of energy by the BS if $V^{init} - V^{pre} > \Delta V_{th}$.

As described in Section 4.5, an advertisement message consisting of battery voltage and location information is transmitted from each sensor node to the BS during the setup phase. The message is sent at the maximum transmission power of the transceiver to guarantee that it can reach the BS. A memory table is maintained at the BS which stores the information of all the nodes such as the ID, the location, the initial battery voltage and the current battery voltage. Whenever an advertisement message arrives, the BS extracts the information of the location and battery voltage of the sender, and updates the memory table. The information in this memory table is used later by the FCM algorithm for the network organization. At the end of the *setup phase*, the BS delivers assignment messages to every sensor node accordingly, this message is also sent at the maximum transmission power of the BS node transceiver.

During the transmission phase, the sensor nodes in a cluster periodically send data packets to their CH at a suitable transmission power calculated and provided by the BS. This transmission power is set based on the distance between the member nodes of the cluster and the CH in such a way that it guarantees at least 90% successful direct transmission between two nodes. As mentioned above, the implementation is performed on IRIS hardware platform equipped with AT86RF230 transceiver. This transceiver allows users to adjust the transmission power among 16 levels from the lowest TX-Level 15 to the highest TX-Level 0 that can help to save energy of the nodes. As mentioned in Chapter 3, Fig. 3.16 obtained by experiment illustrates the success rate of direct data transmission between two nodes when varying transmission power (TX_POWER) at different distances from the sender to the receiver in the line of sight. Since the area of deployment field is $20m \times 20m$, 7 lower levels of the TX_POWER from TX-Level 15 to TX-Level 9 are used when configuring the sensor nodes, TX_POWER is set respective to the transmitting distance as shown in Table 4.2.

Table 4.2: Transmission power level settings with respect to distances for at least 90% successful transmission.

Distance (m)	5	7	8	9	9.5	10	16	20
TX_Power	15	14	13	12	11	10	9	8

4.7 Experimental Results

The performance of the FCMCP is investigated and compared with the well-known cluster-based protocol LEACH Centralized (LEACH-C), which also uses a centralized mechanism of cluster formation. Table 4.3 shows the configuration of the basic network parameters used in the experiment.

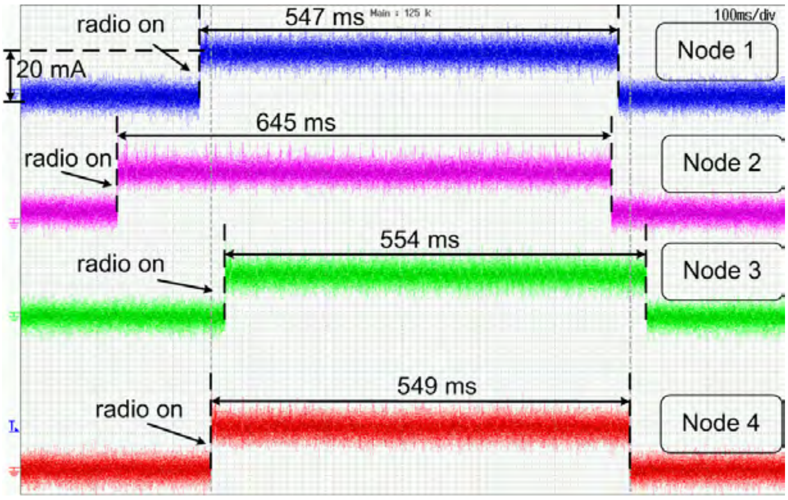
Fig. 4.21a and Fig. 4.21b show the current consumption of the sensor nodes during the *setup phase* and *data transmission phase* respectively in the network of 30 nodes using FCMCP. During the *setup phase*, all the nodes have to turn on their radio component to send the advertisement message and listen to the assignment messages transmitted by the BS. The time interval taken to complete the *setup phase* is in the range of 540 ms to 650 ms for all the sensor nodes. A sample of the *setup phase* at four different nodes is shown in Fig. 4.21a, sensor node 1, 2, 3

Table 4.3: Setting Values for WSN Experiment

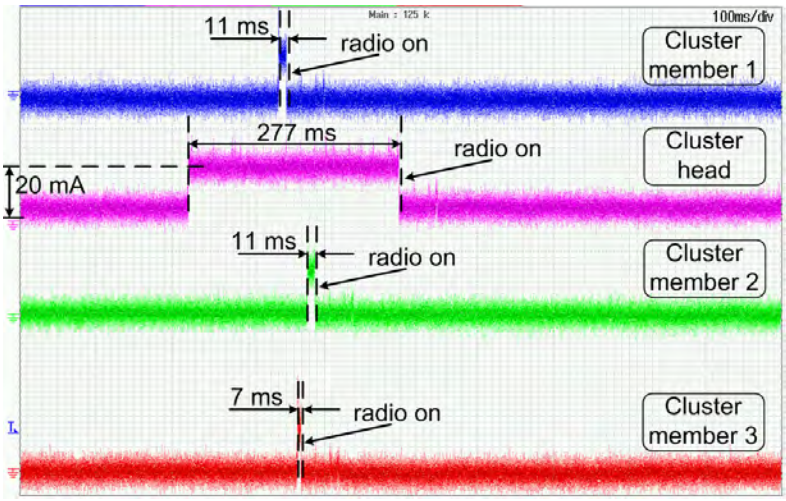
Parameter	Value	Unit
N	30, 40, 50	nodes
c	3	clusters
$T_{ReCluster}$	30,000	ms
$T_{Sensing}$	500	ms
T_{Cycle}	5,000	ms
T_{DataRx}	400	ms
$T_{DataAgg}$	50	ms
$T_{RadioOn\_CH}$	600	ms
$T_{RadioOn\_CM}$	50	ms
ΔV_{th}	100	mV

and 4 take around 547 ms, 645 ms, 554 ms and 549 ms respectively to complete its *setup phase*.

During the *data transmission phase*, the cluster member tries it best to send data before dropping the packet during the transmission time interval, $T_{RadioOn\_CM}$ given at the end of the *setup phase* as shown in Fig. 4.16b. Meanwhile, the maximum time interval a CH turns on its radio is $T_{RadioOn\_CH}$ that includes time for receiving data packets from all the members, processing data and transmit the compressed packet to the BS: $T_{RadioOn\_CH} = T_{DataRx} + T_{DataAgg} + T_{DataTx}$ as illustrated in Fig. 4.16b with the values given in Table 4.3. Fig. 4.21b shows the *data transmission phase* of a cluster containing of 11 member nodes. In this figure, only 3 member nodes of the cluster and its CH are shown. The cluster member nodes 1, 2 and 3 turn their radio on for around 11 ms, 11 ms and 7 ms respectively, meanwhile the time interval of turning on radio component of the CH is around 277 ms for data receiving, processing and transferring to the BS. In the next experiment, the lifetime of the network is studied and compared with other conventional protocols such as LEACH-C and FCMCP. The LEACH-C



(a) Current consumption of the sensor nodes during the *setup phase*.



(b) Current consumption of the cluster member and cluster head during the *data transmission phase*.

Figure 4.21: Measurement of the sensor nodes current consumption in different phases.

is a typical centralized clustering protocol proposed in [47] as an improvement of LEACH. Different from LEACH, the process of CH selection and cluster formation is carried out at the BS instead of performing at the sensor nodes. During the *setup phase*, every sensor node sends the advertisement message including its location and residual energy to the BS. The BS establishes a set of candidates consisting of the nodes with residual energy higher than the average. Simulated annealing algorithm is adopted to find the optimal CHs in such a way that the total distances from cluster members to their CHs is minimized [47]. The objective function is presented in (4.8) as given in [116].

$$f(C) = \sum_{i=1}^{N} min_{c \in C} d^2(i, c) \tag{4.8}$$

where c is the selected CH, C is the set of selected CHs, N is the number of nodes, $d(i, c)$ is distance from node i to CH c. After that, the assignment messages are sent to all the sensor nodes. The *data transmission phase* is performed in the similar way as FCMCP.

To evaluate the performance of each protocol, the following performance parameters are used: t^P_{first} and t^P_{last}, where t^P_{first} is defined as the time until the first node in the network using protocol P runs out of energy and t^P_{last} is the time until the number of alive nodes is smaller than the number of clusters which is three in this case. The comparison of the lifetime between the network using LEACH-C protocol and the FCMCP network is shown in Fig. 4.22 with the 30 nodes and the number of cluster is kept fixed at 3. From Fig. 4.22, it can be seen that by using the FCM algorithm, the lifetime of the network can be prolonged. The time until the first node dies, t_{first} of the FCMCP network is about 16.76 hours, meanwhile when using LEACH-C, the first node runs out of energy after about 12.72 hours. The experiment was run until time t_{last} when the number of alive nodes is smaller than the number of clusters which is three in this case. It also can be observed in Fig. 4.22 that the FCM algorithm enables the network to archive $t^{FCMCP}_{last} = 21.54$ hours, whereas with LEACH-C, the network can obtain $t^{LEACH-C}_{last} = 20.68$ hours which is 4.2% less than t^{FCMCP}_{last}. The reason is that with FCM, a better distribution of the sensor nodes among the clusters can be archived. Therefore, the traffic load via each CH is balanced. By choosing the node with the highest battery voltage as the CH of a cluster, it is guaranteed that the CH has enough energy to receive and transmit data packets. Table 4.4 summarises the investigation of the network

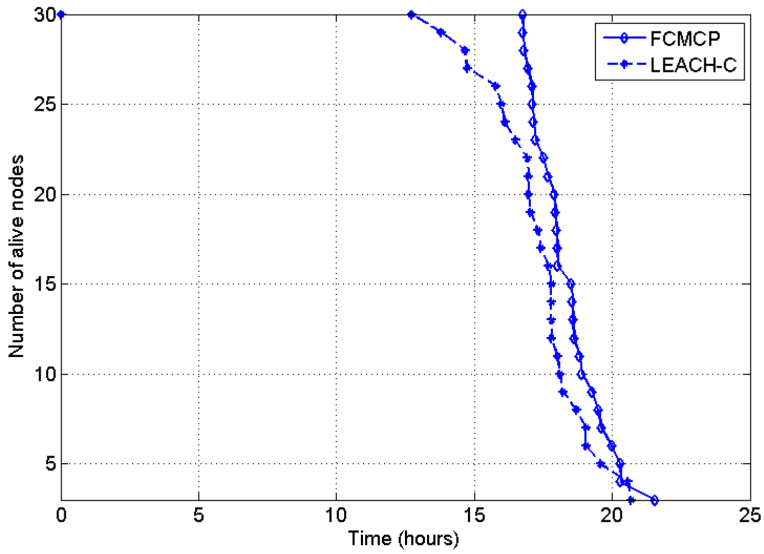


Figure 4.22: Network lifetime with LEACH-C and FCMCP

lifetime, t_{first} as well as the time until last node died, t_{last} with different number of nodes deployed in the area of interest. The results show that the FCMCP networks are able to archive longer lifetime as compared with the LEACH-C networks in all the investigated cases when the number of nodes of the network varies as 30, 40 and 50 nodes. .

Table 4.4: Comparison of network lifetime, t_{first} and t_{last} in hours when the number of sensor node deployed varies

Number of nodes	$t_{first}[hours]$		$t_{last}[hours]$	
	FCMCP	LEACH-C	FCMCP	LEACH-C
30	16.76	12.72	21.54	20.68
40	18.62	16.82	26.74	24.76
50	20.49	18.64	32.30	29.27

4.8 Conclusions

In conclusion, a centralized cluster-based protocol using the FCM algorithm has been investigated by using a simulation tool. It has also been developed, implemented and experimented on a smaller scale WSN test-bed for a real-life fire detection application. The protocol is shown to successfully assist the operation of the WSNs in real-time environment. As shown in the experimental results, it has been proven that the network lifetime of the FCMCP network can be extended when compared with a well-known centralized cluster-based protocol, LEACH-C, the average improvement is around $2 - 4$ hours with number of nodes in the networks varied from 30 to 50 nodes. Furthermore, even though the design and implementation have been carried out for the application of FCM algorithm, the framework can be also used as a test-bed for different optimization algorithms to achieve the best efficiency in WSNs.

Chapter 5

Harmony Search Algorithm based Clustering Protocols

5.1 Introduction

Chapter 4 presents a centralized cluster-based protocol for WSNs using FCM clustering algorithm. The FCMCP enables a better network formation and longer network lifetime when compared with other protocols like LEACH, K-means, etc. However, only the cluster formation in FCM protocol is optimized, the cluster head selection is performed separately within each cluster by choosing the node with highest residual energy. This way of constructing the network may result in not optimal CHs selection among all the sensor nodes. In this chapter, another optimization problem is formulated with the consideration of sensor nodes distribution to cluster and energy efficiency of the CH selection.

Furthermore, the solutions of such a clustering algorithm like FCM are highly sensitive to starting points and frequently converge to local optimum solution or diverge altogether. In that case, evolutionary algorithms may provide a better method to achieve near optimal or global optimal solution. In this chapter, the performance of cluster-based protocols using the conventional methods and evolutionary algorithms such as GA and PSO is investigated and analyzed by simulation. A novel nature-inspired optimization algorithm, called Harmony Search Algorithm (HSA) is adopted to be integrated into a cluster-based protocol so that a better network configuration is achieved. The performance of this protocol is evaluated by simulation and then realized and validated on a hardware test-bed.

5.2 Optimization Problem of the WSNs

A WSN of N sensor nodes is randomly deployed into a field with an area $M \times M\, m^2$ and organized into c clusters : $C_1, C_2, ..., C_c$. The working conditions of the sensor nodes as well as the network are considered the same as the assumptions presented in Chapter 4. In the proposed centralized cluster-based protocol for this WSN, the BS needs to select the CHs with higher residual energy among the sensor nodes, and then forms the clusters with equal distribution of the sensor nodes based on their information of location and residual energy. This process can be formulated as an optimization problem and mathematically expressed as shown in Equation

(5.1):

$$f_{obj} = \alpha \times f_1 + (1-\alpha) \times f_2 \tag{5.1}$$

where

$$f_1 = max_{j\in(1,c)}\left\{\frac{\sum_{\forall node_i \in C_j} d(\text{node}_i, CH_j)}{\|C_j\|}\right\}$$

$$f_2 = \sum_{j=1}^{c}\left\{\frac{\sum_{i=1}^{\|C_j\|}\Delta V_i}{\Delta V_{CH_j}}\right\}$$

By minimizing the objective function, f_{obj}, it is expected that the cluster formation and the CH selection of the WSN can be optimized for increasing the efficiency of distributing energy consumption within the network. As presented in (5.1), f_{obj} consists of two parts. The first part, f_1 is the maximum of the total of the Euclidean distance of the nodes, $\text{node}_i\ \forall i \in$ cluster C_j, to their CHs CH_j and $|C_j|$ is the number of nodes that belong to cluster C_j. By minimizing f_1, it tends to minimize the intra-cluster mean distance between these sensor nodes and their respective CHs. Meanwhile, the second part, f_2 is the sum of the ratio of present residual energy of all alive nodes, $\sum_{i=1}^{|C_j|}\Delta V_i$ in the network with the present energy level of the CH_j, which is ΔV_{CH_j}, in the present round. In a practical system where each sensor node is powered by batteries, the residual energy of node i can be represented by its battery terminal voltage, i.e. $\Delta V_i = (V_i^{init} - V_i^{pre})$, where V_i^{init} and V_i^{pre}) are the initial and present voltage of node i respectively. In order to minimize f_2, the $\sum_{i=1}^{|C_j|}\Delta V_i$ need to be minimal, while ΔV_{CH_j} must be maximized. The sensor node with higher energy level within each cluster tends to be the CH. Hence, minimizing f_2 results in the selection of the optimal CHs in terms of energy in the network.

The constant α indicates the contribution of f_1 and f_2 in the objective function f_{obj}. In order to get rid of the selection of the nodes with low energy level to be the CH, the CH nodes are selected in the set of candidates: $\{\text{node}_i \mid \Delta V_i \geq \Delta V_{ave}\}$. Only the sensor nodes which has the energy level higher than the average energy level of all the nodes, ΔV_{ave} in the network can be the CH candidates.

5.3 Harmony Search Algorithm

In this section, an engineering approach applied in the design and development of an algorithm inspired from musical process of searching for a perfect state of harmony [23] is introduced. Harmony search algorithm has been very successful in a wide variety of optimization problems, presenting several advantages with respect to traditional optimization technique [24, 25]. HSA imposes fewer mathematical requirements and does not require initial value settings of the decision variables. In the HSA, musical performances seek a perfect state of harmony determined by aesthetic estimation, as the optimization algorithms seek a best state (i.e. global optimum) determined by objective function value. Various steps involved in a HSA-based approach for minimizing the objective function (5.1) have been explained, describing how the HSA is designed and applied to the present problem.

Step 1: *Initialize the optimization problem and algorithm parameters.*

For the present problem, we have to minimize the intra-cluster distance and optimize the energy consumption of the network, which is defined by the objective function as given in (5.1). The solution of the formulated problem is the index of the k cluster heads among the candidate nodes in the network. The number of solution vectors in Harmony Memory Matrix is the Harmony Memory Size (HMS).

It is similar to population in GA. For the formulated problem we considered HMS as 50. In order to use this memory effectively; the HSA adopts a parameter called Harmony Memory Considering Rate (HMCR). If this rate is too low, only few elite harmonies are selected and it may converge too slowly. If this rate is extremely high (near 1) then, the pitches stored in the harmony memory are mostly used, and newer pitches are not explored well. Therefore, we assumed HMCR as 0.95. The second component is the pitch adjustment which is governed by Pitch Adjusting Rate (PAR), which is assumed as 0.7. The maximum number of searches (stopping criterion) is selected as 100. HMCR, PAR, and the maximum number of searches (stopping criterion) initialized are used to improve the solution vector.

Step 2: *Initialize the Harmony Memory (HM)*

The initial HM consists of a HMS number of randomly generated solutions for the optimization problem under consideration. The solution of the formulated problem is identification (ID) of the N numbers of cluster heads. HM with the size of HMS can be represented by (5.2).

$$HM = \begin{bmatrix} I_1^1 & I_2^1 & \cdots & I_k^1 \\ I_1^2 & I_2^2 & \cdots & I_k^2 \\ \vdots & \vdots & \ddots & \vdots \\ I_1^{HMS} & I_2^{HMS} & \cdots & I_k^{HMS} \end{bmatrix} \begin{bmatrix} F^1 \\ F^2 \\ \vdots \\ F^{HMS} \end{bmatrix} \tag{5.2}$$

Each row of the HM is a random solution for the optimization problem and then the value of the objective function given by Equation (5.1) is computed for each harmony vector and represented by F^j.

Step 3: *Improvise a new harmony from the HM*

After defining the HM as shown in (5.2), the improvisation of the HM is done by generating a new harmony vector $\begin{bmatrix} I_1' & I_2' & \cdots & I_k' \end{bmatrix}$. Each component of the new harmony vector I_j' is generated using (5.3) based upon the HMCR defined in Step 1.

$$I_j' \leftarrow \begin{cases} I_j' \in HM \text{with probability HMCR} \\ I_j' \in I_j \text{with probability (1-HMCR)} \end{cases} \tag{5.3}$$

HMCR is defined as the probability of selecting a component from the HM members, and (1-HMCR) is, therefore, the probability of generating it randomly. If I_j' is generated from the HM,then it is further modified or mutated according to PAR. The PAR determines the probability of a candidate from the HM to be mutated and (1-PAR) is the probability of doing nothing. The Pitch adjustment for the selected I_j' is given by (5.4)

$$I_j' \leftarrow \begin{cases} I_j^n \in HM \text{with probability PAR} \\ I_j' \text{with probability (1-PAR)} \end{cases} \tag{5.4}$$

Where I_j^n is nearest node whose energy is greater than the average residual energy.

Step 4: *Update the HM*

The newly generated Harmony vector is evaluated in terms of the objective function value. If the objective function value for the New Harmony vector is better than the objective function value for the worst harmony in the HM then new Harmony is included in the HM and the existing worst harmony is excluded from the HM.

Step 5: *Go to step 3 until termination criterion is reached.*

The current best solution is selected from the HM after the termination criterion is satisfied. This is the solution for the optimization problem formulated.

5.4 A Harmony Search Algorithm based Clustering Protocol

The operation of the protocol includes 2 phases: *clustering setup* and *data transmission*. In *clustering setup* phase, sensor nodes are grouped into clusters in such a way that the objective function (5.1) is optimized. Then, the cluster heads collect data from all cluster members and transfer to the BS during the *data transmission*. The two phases are performed in each round of the network operation and is repeated periodically.

Phase 1: Clustering setup

In the clustering setup phase, these sensor nodes send an ADVERTISEMENT (ADV) messages to the BS with the information of their geographical location and battery voltage; this information is later forwarded by the BS node to the computer for calculation. After finding the optimum and organizing the network into clusters, the BS attaches the information of the cluster head and to which cluster the normal nodes belong into the ASSIGNMENT (ASG) messages and transmits this message to every sensor nodes. The network can be clustered as shown in Algorithm 6.

After the clusters are created, the data transmission phase is performed that allows the cluster members of each cluster to send data towards the BS through the CHs. The process of selecting clusters is repeated periodically whenever the

Algorithm 6 Cluster formation

repeat
  Initialize k randomly selected cluster heads
  for i = 1 to N **do**
    Calculate the distance $d(node_i, CH_j)$ between $node_i$ and all cluster heads CH_j.
    Assign $node_i$ to the cluster C_j in which $d(node_i, CH_j)$ is minimum
  end for
  Compute the value of the objective function given in Equation (5.1)
  Update the cluster head by using HSA
until The maximum number of iteration is reached

data transmission phase is completed.

Phase 2: Data transmission

After all the nodes receive the ASG message, and the transmission schedule is initialized, the sensor nodes start to perform sensing task and transmit data to the CHs. Transmission power of cluster member nodes is optimized because of the minimum spatial distance to the CHs. A loosen time synchronization is created amongst the nodes of each cluster. The CH radio component is turned on first at time t_1^{CH}, and it remains on until all the cluster members' data packets are received or a predefined listening period of time is out. Then it is turned off at time t_2^{CH}. Meanwhile, cluster member nodes turn on their radio component for a very short period of time which is (t_1^{CM}, t_2^{CM}) in order to transmit data. This period of time is guaranteed that $t_1^{CH} < t_1^{CM}$ and $t_2^{CH} < t_2^{CM}$ for every cluster member in the cluster. Data aggregation and fusion are carried out at the CHs, and only the compressed data packet is sent from the CHs to the BS, thus the amount of information transmission is reduced that results in the reduction of energy consumption.

5.5 Simulation Results and Discussions

The evaluation of the algorithm is fulfilled by simulating a 100 node network with MATLAB. Sensor nodes are deployed in an area of $100 \times 100\, m^2$ and the initial energy of each node is 1 J. The base station is located at the position $(50, -75)$ and the number of clusters is $k = 5$. The coefficient α in (5.1) is set as $\alpha = 0.5$ as given in [31].

5.5.1 Convergence comparison

The HSA has been applied for finding the optimal cluster heads for the WSN. The parameters of HSA such as the HMS, HMCR and the PAR were selected as 50, 0.95 and 0.8 respectively. Evolutionary algorithms such as GA and PSO are also employed to solve the formulated optimization problem. Since Evolutionary algorithms are heuristic in nature, so, we performed 50 trials to obtain the best solution. The convergence of the best fitness value for the objective function given by (5.1) with the number of generations using GA, PSO and HSA are shown in Fig. 5.1.

In Fig. 5.1, it can be seen that HSA converges at a much faster rate and achieves a better minimum fitness value for the objective function as compared to PSO and GA. Lack of diversification in GA reduces its search capability and leads to premature convergence and that leads to a higher probability toward obtaining a local optimum [85]. PSOs main drawback is its inability to maintain the desired levels of population diversity and the balance between local and global searches

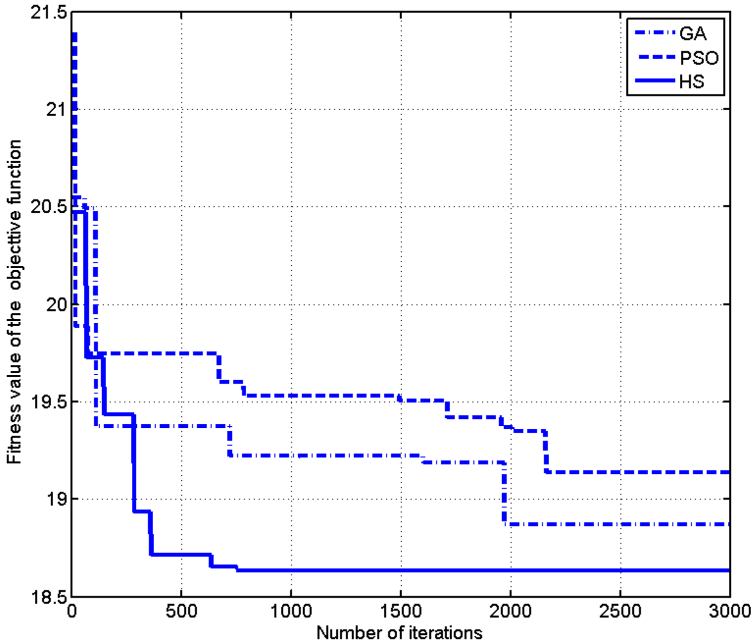


Figure 5.1: Convergence of the objective function

and hence suboptimal solutions are prematurely obtained [89]. In general, the randomization character of evolutionary algorithms adds a degree of diversity to the manipulated populations. Nevertheless, in PSO these random components are unable to add a sufficient amount of diversity. As shown in [90] frequent collisions of particles in the search space, especially onto the leader, can be detected. This, in fact, causes the effective population size to be lower than actual and consequently reduces the effectiveness of the algorithm. Also the PSO algorithm includes some tuning parameters that greatly influence the algorithm performance. Whereas in the HS algorithm the randomization explores the search space more widely and efficiently; the pitch adjustment ensures that the newly generated solution is good enough, or not too far from existing good solutions. Therefore, the randomization diversifies the search space and helps HS coming out of the local minima and pitch adjustment intensifies the convergence to the value surrounding the global minima.

5.5.2 Network performance

In Fig. 5.2, the number of alive nodes over the operating time of the network by using different protocols is compared. The performance of LEACH, K-means, FCMCP, GA, PSO and Harmony Search Algorithm based Clustering Protocol (HSACP) are studied. The different duration of time up to the first dead node by applying different protocols is given in Table 5.1. It is obviously seen that the lifetime of the network with 100% nodes alive when using clustering protocol with HSA, PSO, GA, K-means and FCM is much longer than the network lifetime shown in Table 5.1 when LEACH is employed. Those four algorithms produce better clustering structure of the network and the cluster heads are distributed more uniformly across the network. The energy consumption of all nodes are reduced because of shorter distances between non-cluster head nodes and their cluster heads. With HSA, PSO and GA, the selection of cluster head is fulfilled by using a cost function including nodes' residual energy which assists to find the optimal cluster head. Meanwhile, with K-means and FCM, the clusters are formed without considering residual energy, and cluster heads are chosen randomly among the nodes in the clusters.

Table 5.1: Duration of time up to the first node dies in the network

Protocols	Lifetime of the first dead node (rounds)
LEACH	1473
K-Means	1608
FCMCP	1701
GA	1758
PSO	1807
HSACP	1863

The simulation is run to observe the changing of energy consumption with

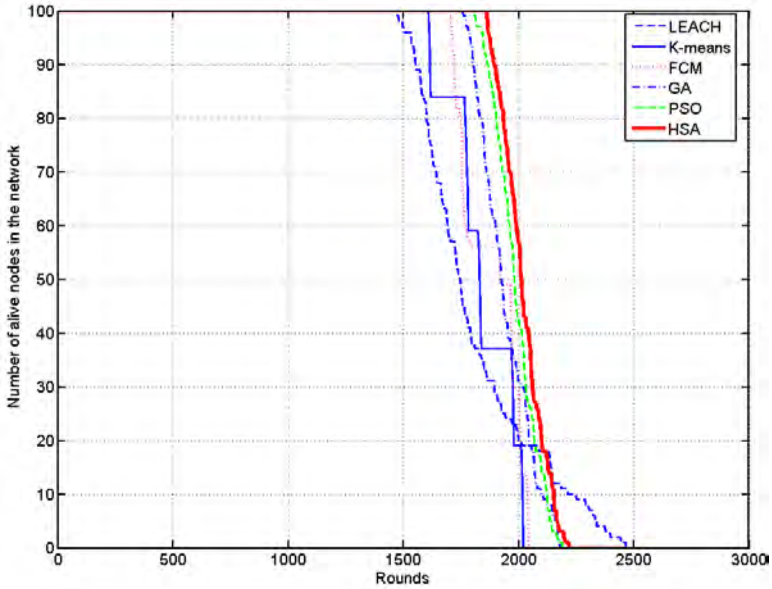


Figure 5.2: Number of alive node vs. time

the variation of network diameter. Fig. 5.3 shows the average energy consumption of the network with different algorithm over the diameter after 200 rounds. With the small network diameter, energy consumption of the network is almost the same when using different protocols. However, when the network diameter increases, the average energy consumption of all nodes using HSACP is better than that of other methods. The reason is that the optimal cluster distribution among the network.

5.6 Real-time Implementation of the HSACP for WSN

The HSACP is realized in real-life application using WSNs to monitor the environment. The layer at the node level is designed and developed with the support of the TinyOS operating system 2.1.1. Meanwhile the BS level, where the HSA is performed, is written in Java language and run on a Linux based computer. The framework of the hardware platform is based on that described in Fig. 4.14 in

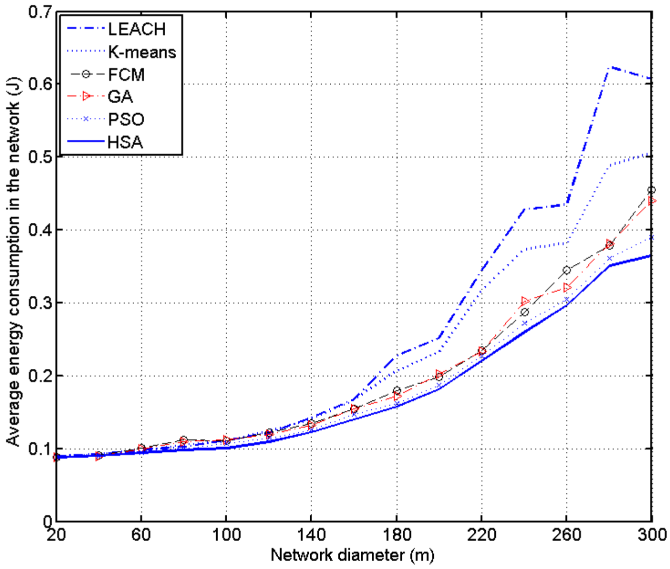
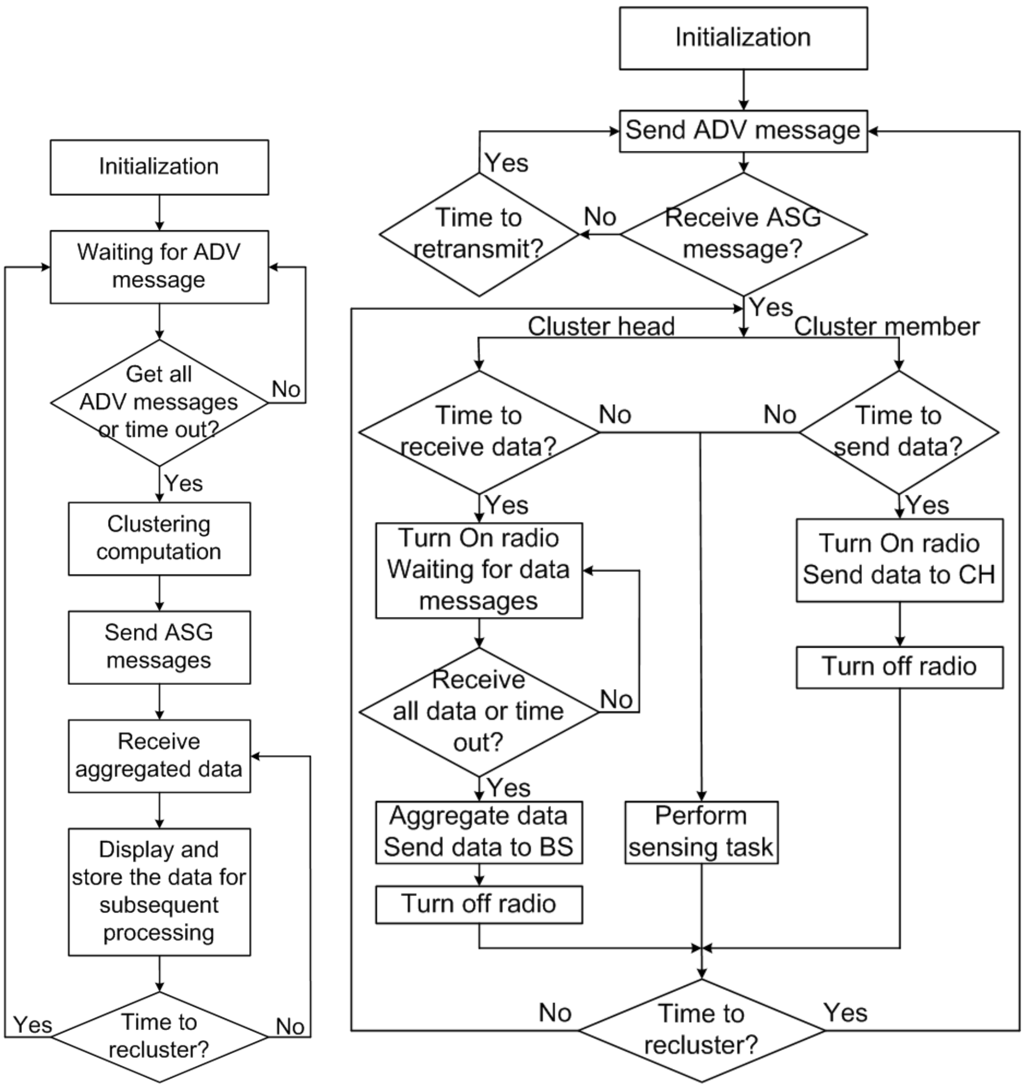


Figure 5.3: Average energy consumption of different protocols

Chapter 4. The proposed HSACP uss the same structure of hardware platform illustrated in Fig. 4.15. The HSA is integrated in the *Clustering Algorithms* module of the **Cluster Management** at the BS level.

The operation of the network protocol at the BS and sensor node is shown in the flowchart in Fig. 5.4. At the beginning of the operation cycle, all the sensor nodes in the network send an ADV message to the BS and keep the radio component on to wait for the ASG message. Once, the ASG message is received, the sensor node extracts the information to identify whether it is a CH or a cluster member to set the operation of the data phase accordingly. The structure of the ASG message is provided in Fig. 4.18. The cluster members simply send the data to their CH after performing the sensing task. Meanwhile, the CH needs to collect data from its members, aggregate with its own measurement, and then send the compressed data packet to the BS before turning off the radio component as shown in Fig. 5.4.



(a) Base station.

(b) Sensor nodes.

Figure 5.4: The flowchart of the network operation at the BS and sensor nodes.

The BS takes the responsibility to control the formation of the network and acquire sensing data from the nodes. During the setup phase, the BS collects the information of each sensor node when receiving the ADV message which has the structure as shown in Fig. 4.17 . It includes the location and battery voltage of the sensor nodes. The sensor node information received at the BS node is forwarded to the computer via a serial port. Once the ADV messages from all the nodes are collected, the HSA algorithm is run at the computer to select the CH for each cluster and calculate the cluster formation. When the computation is completed, the BS node sends assignment messages to every node in the network which contains the information of the CH as well as the time schedule for data transmission and cluster re-organization. The data acquisition is later started to obtain and display the sensing measurements of the network.

5.7 Experimental Results

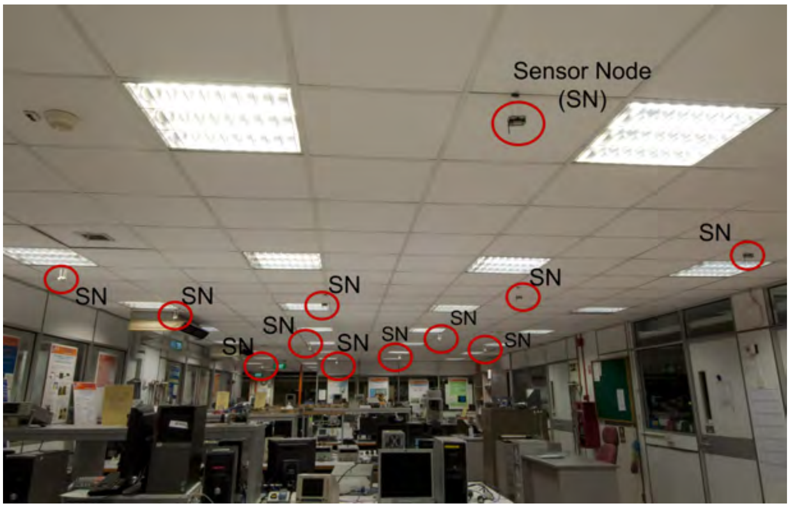
5.7.1 Experiment Setup

Experiments are conducted for a WSN of 30 nodes deployed as illustrated in Fig. 5.5a. The sensor nodes form the cluster-based network, perceive ambient temperature and send the measurements to their CHs. The CHs aggregate these information and transmit the compressed data packets to the BS. Network formation and monitored data are displayed on a Graphic User Interface (GUI) as shown in Fig. 5.5b.

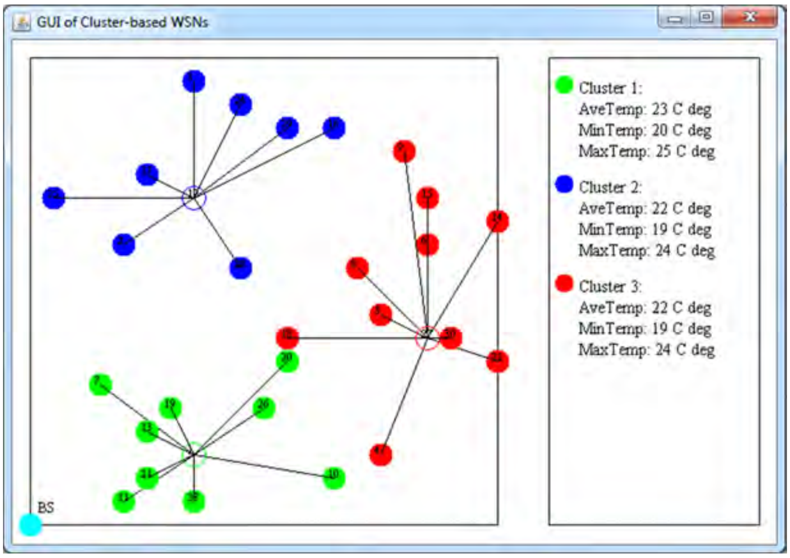
The setting values installed for network configuration during the experiments are provided in Table 5.2.

5.7.2 Investigation of the convergence and computational time

As mentioned above, f_{obj} consists of two parts: f_1 and f_2, which have the contribution to f_{obj} defined by the coefficient α. By setting α to 1 and 0, then executing the algorithm for a 30-node WSN, it is found that the values of f_1 and f_2 is in



(a) Experiment setup of the WSN.



(b) The GUI displays the topology of the network and sensing measurements.

Figure 5.5: Experiment setup and visualization.

the range of $228.14 \div 262.23$ and the $14.89 \div 16.23$. Therefore, $f_2 \approx 0.06 f_1$. The value of f_1 is large as compared to that of f_2, therefore, f_1 is dominant in the formulated objective function f_{obj}. Thereby, energy efficiency of the CH selection decided by f_2 might be reduced. In order to avoid this issue, the coefficient α in (5.1) is set as small as $\alpha = 0.05$ which gives an equal contribution of f_1 and f_2, i.e. $\alpha f_1 \approx (1-\alpha) f_2$. The HSA is then adopted for finding the optimal cluster

Table 5.2: Setting Values for WSN Experiment

Parameter	Value	Unit	Description
N	30	nodes	Total number of nodes
c	3	clusters	Number of clusters
$T_{ReCluster}$	30,000	ms	Time to recluster
$T_{Sensing}$	500	ms	Sample time for sensing
T_{Cycle}	5,000	ms	Time interval between two data transmission
T_{DataRx}	600	ms	Time to receive data of the CH
$T_{DataAgg}$	50	ms	Time to aggregate data at the CH
$T_{RadioOn\_CH}$	600	ms	Maximum time to keep radio on for sending one data packet at the CH
$T_{RadioOn\_CM}$	50	ms	Maximum time to keep radio on for sending one data packet at the CM
ΔV_{th}	100	mV	Voltage threshold for dead node

heads for the WSN. The parameters of HSA such as the HMS, HMCR and the PAR are selected as 50, 0.95 and 0.7 respectively. Since evolutionary algorithms are heuristic in nature, so, we performed 50 trials to obtain the best solution. The convergence of the best fitness value for the objective function given by (5.1) over the number of generations using HSA during the experiments is shown in Fig. 5.6. It can be seen that fitness value can converge after less than 30 iterations that is fast enough for real time network configuration.

The computational time of the algorithm as well the time taken by different protocols for completing the setup phase are shown in the Table 5.3. The study of these periods of time is conducted with a 30-node network, the calculation is made for 50 cycles of reorganizing the network and average values are provided. As observed from these values, the computational time of the algorithms FCM and HSA are insignificant when compared with the time duration of the setup phase. Therefore, it enables these algorithms to be applied for online configuration

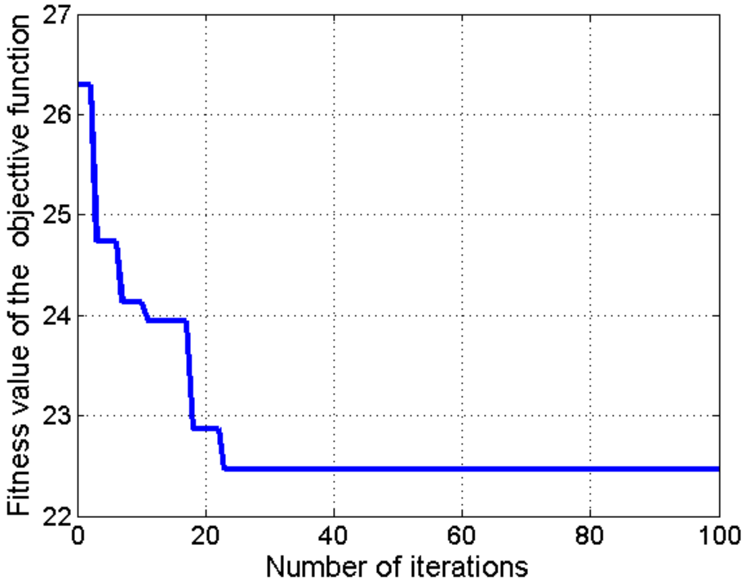


Figure 5.6: The convergence of the objective function when using HSA

of the network. In addition, the setup phase durations of all the three studied protocols are almost the same. It means that the cost of optimization process by using HSA is comparable with the cost of setting up the network done by the other two methods. Hence, the difference of the network performance with these protocols only represents the achievement of the optimization process during the data transmission phase.

Table 5.3: Computational time and setup phase duration

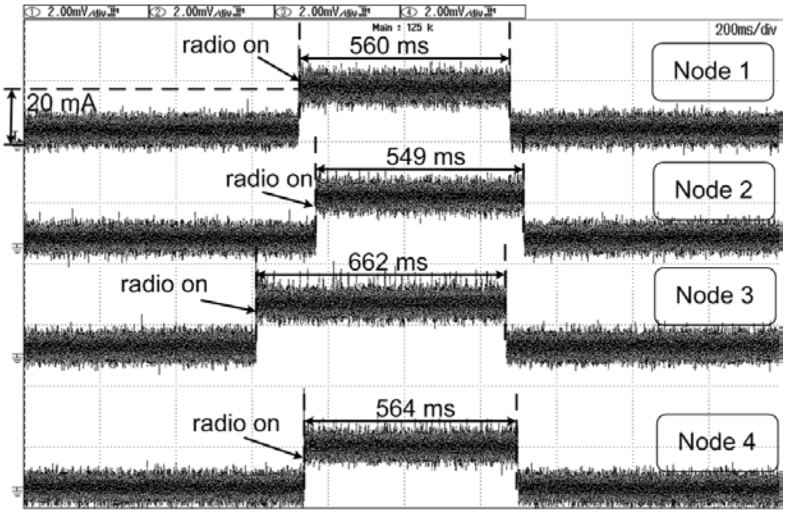
Protocols	LEACH-C	FCMCP	HSACP
Computational time (ms)	12	3	8
Setup phase duration (ms)	1194.73	1152.37	1153.56

5.7.3 Experimental results of the network performance

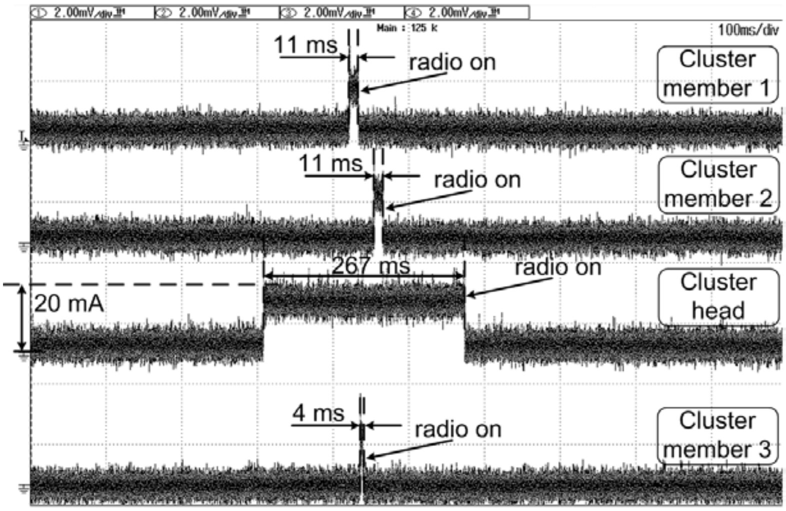
First, the operation of the real-time networking system using HSACP is illustrated in Figs. 5.7a and 5.7b. Figs. 5.7a and 5.7b show the current consumption of the sensor nodes during the *setup phase* and *data transmission phase* respectively in the network of 30 nodes using HSA protocol. During the *setup phase*, all the nodes have to turn on their radio component to send the advertisement message and listen to the assignment messages transmitted from the BS. The time interval taken to complete the *setup phase* is in the range of 540 ms to 700 for all the sensor nodes. A sample of the *setup phase* at four different nodes is shown in Fig. 5.7a, sensor node 1, 2, 3 and 4 spend around 560 ms, 549 ms, 662 ms and 564 ms respectively to complete its *setup phase* with node 3 being the longest.

During the *data transmission phase*, the cluster member tries it best to send data before dropping the packet during the transmission time interval, $T_{RadioOn\_CM}$ given at the end of the *setup phase*. Meanwhile, the maximum time interval a CH turns on its radio is $T_{RadioOn\_CH}$ that includes time for receiving data packets from all the members, processing data and transmitting the compressed packet to the BS: $T_{RadioOn\_CH} = T_{DataRx} + T_{DataAgg} + T_{DataTx}$ with the values given in Table 5.2. Fig. 5.7b shows the *data transmission phase* of a cluster containing of 12 member nodes. In this figure, only 3 member nodes in this cluster and its CH are shown. The cluster member nodes 1, 2 and 3 turn their radio on for around 11 ms, 11 ms and 4 ms respectively, meanwhile the time interval of turning on radio component of the CH is around 267 ms for data receiving, processing and transferring to the BS.

In the next experiment, the lifetime of the network is studied and compared with other conventional protocols such as LEACH-C and FCMCP presented in Chapter 4.



(a) Current consumption of the sensor nodes during the *setup phase*.



(b) Current consumption of the cluster members and CH during the *data transmission phase*.

Figure 5.7: Measurement of the sensor nodes current consumption in different phases.

The comparison of the lifetime amongst the networks using a LEACH-like protocol, FCMCP and HSACP is shown in Fig. 5.8. As mentioned in Chapter 4, t^P_{first} is defined as the time until the first node in the network using protocol P runs out of energy and t^P_{last} is the time until the number of alive nodes is smaller than the number of clusters which is three in this case.

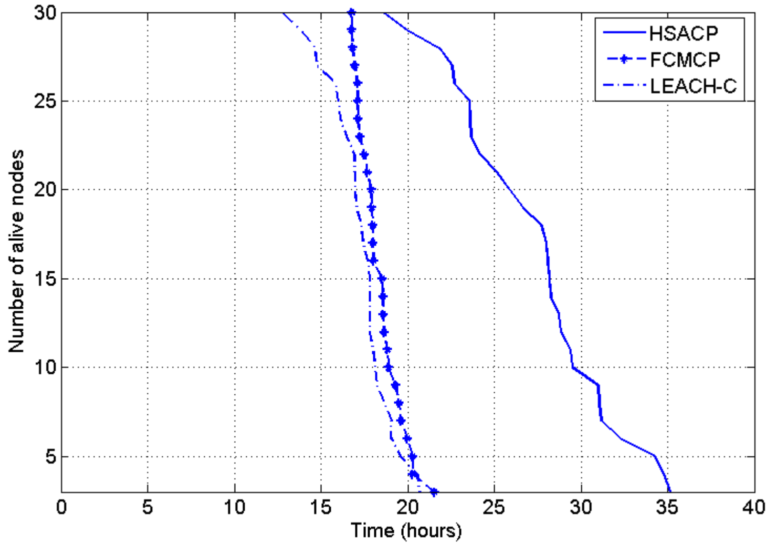


Figure 5.8: Comparison of the network lifetime with LEACH and FCMCP

It is observed in Fig. 5.8 that the network with LEACH-like protocol has $t_{first}^{LEACH-C} = 12.72$ hours, meanwhile in the FCMCP network, $t_{first}^{FCMCP} = 16.76$ hours, and in the HSACP network $t_{first}^{FCMCP} = 18.56$ hours that is the longest period of time. As mentioned above, LEACH chooses the CHs in a random way. Although, it attempts to rotate the CH role evenly among the sensor nodes by using a predefined probability of being CH, it is not guaranteed the equal distribution of the nodes into clusters as well as some nodes can be selected as the CH for more times. Hence, the energy capacity is drained faster. Meanwhile, FCMCP assists the network to be organized in a better way, sensor nodes are evenly allocated into clusters. Therefore, the traffic load at the CHs can be balanced amongst the clusters and fast depletion of energy at the CH can be avoided. However, the CH is only chosen among the nodes within a cluster that may not be the best in the network in term of energy distribution. When HSACP is applied, the network formation and cluster head selection are considered at the same time in order to further optimize the network organization. The best CHs in terms of energy efficiency can be selected, whereas clusters can be formed optimally. The efficiency of HSACP can be seen clearer by comparing the time t_{last}^{P}. Fig. 5.8 shows that the time $t_{last}^{HSACP} = 35.17$ hours that is much longer than that of FCMCP, $t_{last}^{FCMCP} = 21.54$ hours and that of LEACH, $t_{last}^{LEACH-C} = 20.68$ hours. At the beginning, all the nodes have the same energy level. However, during the network operation, at various time instance, each node has a different residual energy. LEACH-C and FCMCP algorithms focus only on obtaining uniform distribution of the sensor nodes into different clusters; the selection of the CHs for energy efficient operation is not optimized. It is therefore, the network lifetime may not have been maximized. Meanwhile, HSACP involves a better way of optimizing the network control, thus, better performance presented by longer network lifetime is achieved.

5.8 Conclusions

In this chapter, a centralized cluster-based protocol is presented. This protocol uses HSA to create cluster structure in order to minimize the intra-cluster distance and optimize the energy consumption of the network. With the facilitation of data aggregation and cluster head rotation, energy consumption is balanced among all the sensor nodes and the amount of data transmitted to the BS is reduced remarkably. The performance of the protocol has been first investigated in simulation. It is shown that it is able to improve the network performance by using HSA when compared with other evolutionary algorithms such as GA and PSO. The proposed protocol has been successfully implemented practically for a real-life application. The experimental results show that by using HSACP, the network lifetime is extended significantly when compared with LEACH-C and FCMCP. It is clear from the result that HSA can provides fast convergence with the best fitness value;

computational time is insignificant that enables HSA to be applied for online configuration of the network. Generally, it can be concluded that the HSA's simplicity of implementation, high quality solution, along with the lower number of setting parameters makes it an ideal method when dealing with complex engineering optimization problems.

Chapter 6

Energy Efficient Multihop Communication for Large Scale WSNs using Intelligent Water Drops Algorithm

6.1 Introduction

The *cluster-based protocols* such as FCMCP and HSACP presented in Chapter 4 and Chapter 5 respectively, group sensor nodes into clusters, utilize data aggregation technique at the cluster head level to gather data from cluster members, then they send the compressed data to the BS. These protocols premise that every sensor node can send packets to the BS in single hop. This assumption limits the size of the network as the limitation of sensor nodes communication range. In addition, it restricts the controllability of the transmission power, since the further nodes have to consume more energy to transfer packet to the BS.

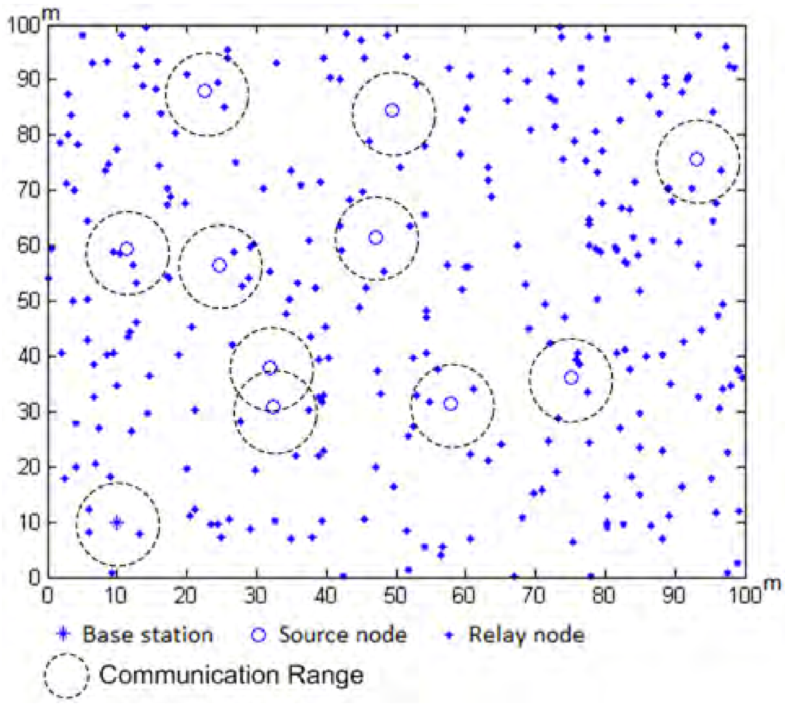
In this chapter, the second category of the network architecture, i.e. the *tree-based protocols*, are focused. These protocols organize sensor nodes into a tree where data aggregation is performed at intermediate nodes, which are located at the junctions of tree branches. The aggregated data packets are later routed to the root node, i. e. the BS. The tree-based protocols are suitable for applications, which involve in-network data aggregation [19]. Examples of the applications can be forest fire detection, safety monitoring in industrial plants, etc., where the measurement provides the most useful information about the safety conditions. One of the main objectives of the tree-based protocols is to optimize the construction of a data aggregation tree in terms of energy efficiency. This optimal aggregation tree is recognized as a NP-Hard problem [96], which is equivalent to Steiner tree, weighted set cover problem [97]. A novel nature-inspired optimization method, called Intelligent Water Drop (IWD) [27], is adopted with the expectation that the optimal or near optimal tree is achieved. The method mimics the dynamics of a river system and the behaviour of water drops when they are moving in the river such as the variation of velocity, the change of soil in the bed of the river, the change of the flow direction, etc. The IWD algorithm changes the amount of soil on the paths, that water drops traversed. This variation depends on the velocity and the soil carried by the water drop, and it can be increased or decreased to attract or obstruct other water drops. During this process, the optimum solution is gradually approached, in this problem, it is the optimal data aggregation tree.

The basic IWD algorithm forms the aggregation tree with the re-enforcement of attracting other water drops in the next round of the tree construction at the junction nodes of the tree. However, this has limitation when no junction node is found during the earlier rounds. Thus, a modification of the basic model IWD algorithm is proposed in this chapter in order to effectively apply the method for solving data aggregation optimization problem. For the purpose of enhancing the probability of finding the destination or aggregation node, the attraction for new IWDs is performed at all nodes that have been visited by an IWD along the route from one source to the destination, i.e. the BS.

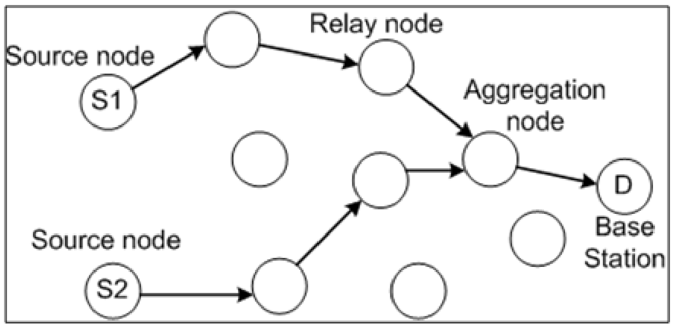
6.2 Optimization Problem and Network Assumptions

A WSN with a number of sensor nodes randomly and densely deployed in an area is considered. It is assumed that every sensor node, which has a unique identity, is stationary and powered by a finite energy source with the same initial energy capacity. There are three types of sensor nodes defined in this network: *source nodes*, *relay nodes* and a *BS*. Each node can send packets to its neighboring nodes within its limited communication range as shown in Fig. 6.1a. Whenever a source node needs to send data to the BS, it has to find a route formed by a set of relay nodes to transfer data. The relay node, that receives data from different sources or is at the junction of different routes, is an aggregation node as illustrated in Fig. 6.1b. This node gathers data, combines all of the packets received into a single packet and then forwards the concise data packet to the next relay node or the BS.

In order to save the total energy consumption of the WSN, the number of data packets relayed in the network needs to be reduced. This means the number hops on the routes from the sources to the destination should be minimized. Furthermore, the junction nodes of these routes need to be as near as possible to the sources in order to perform data aggregation at an early stage of transfer-



(a) Deployment of sensor nodes.



(b) Data aggregation scheme.

Figure 6.1: Deployment of sensor nodes in the environment field and the data aggregation scheme within the network.

ring data. The WSN can be represented as a directed graph $G = (V, E)$, where V is the set of sensor nodes with $|V| = n$ is the number of sensor nodes, and $E = \{(u, v)\,|(u, v \in V), d(u, v) \leq R)\}$ denotes the set of edges connected nodes in their mutual neighborhood, $d(u, v)$ is the Euclidean distance between node u and node v, and R is the communication range of every node in V. The set $S = \{s \in V\}$ is defined as the group of source nodes that have data to send in the network, and the node b is the destination for the data to reach, which is the base station. The optimization problem mentioned above can be defined as: finding a subset $G_{agg} = (V_{agg}, E_{agg}) \subset G$ to

$$minimize \sum_{s \in S,\, v \in V_{agg}} (h_{sv} + h_{vb})$$

where h_{sv} and h_{vb} are the number of hop count from source node, s to aggregation node, v, and from node v to the destination b respectively; $V_{agg} \supset S$, b.

In other words, if the length of the route is calculated as the number of edges, the objective of this optimization problem is to minimize the number of edges connected nodes from all sources to the destination in the network. By constructing such an optimal data aggregation tree, the total energy consumption of transmitting data within the network can be minimized.

In this study, the data packet loss during the transmission is not considered, thus energy wasted for losing data packets is not taken into account. The time of network operation is discrete and defined by the number of runs. During each run, when a data aggregation tree is constructed, all the sources transfer data to the base station through this tree. In order to evaluate the energy efficiency of the proposed algorithm, the first-order energy model presented in [18] is used to compute the total energy consumption of the network. E_{Tx} consumed to transmit a l-bit message over a distance d is calculated by using the Equation 3.1. While to receive this message, the energy consumption, E_{Rx} is calculated by using the Equation 3.3

6.3 Principles of Intelligent Water Drops Algorithm

In large sensor networks, finding optimal aggregation tree is NP-hard problem. To solve this problem, intelligent water drops algorithm, a nature inspired optimization method, is adopted. This algorithm is inspired by the observation of natural water flow in the rivers formed by a swarm of water drops [27]. The swarm of water drops find their own way to the lakes or oceans while it has to overcome numbers of obstacles in its path. Without the presence of these obstacles, the water drops tend to be pulled straight towards the destination by the gravitational force. However, being blocked by different kinds of obstacles and constraints, there exist lots of twists and turns in the real path of the river. The interesting point is that the path of the river, constructed by the flow of water drops, seems to be optimized in terms of distance from the source to the destination under the constraints of the environment. By mimicking the features of water drops and obstacles of the environment, the IWD algorithm uses a population of water drops to construct paths and obtain the optimal or near optimal path among all these paths over time. The environment represents the optimization problem needed to be solved. A river of IWDs looks for an optimal route for the given problem.

Hosseini presented the basics of IWD algorithm [28], then applied it to solve different optimization problems. As described in [28], an IWD model is proposed with two important parameters:

- The amount of soil it carries or its soil load, $soilLoad^{IWD}$).

- The velocity at which it is moving, vel^{IWD}.

The values of these two parameters may change as the IWD flows in its environment from the source to a destination. An IWD moves in discrete finite-length steps and updates its velocity by an amount Δvel^{IWD} when it changes the position from point i to point j as follows.

$$\Delta vel^{IWD} = \frac{a_v}{b_v + c_v.soil^2(i,j)} \tag{6.1}$$

where $soil(i,j)$ is the soil on the bed of the edge between two points i and j; a_v, b_v and c_v are pre-defined positive parameters. The relationship between velocity and the amount of soil of the edge is decided by a_v and c_v, meanwhile b_v is a small number used to prevent the singularity problem. Equation (6.1) indicates that the rate of changing the velocity is dependent on the soil of the edge, i.e, edge with more soil provides more resistance to the water flow that results in the smaller increment in velocity and vice versa. Thus, the velocity at time $(t+1)$ is

$$vel^{IWD}_{(t+1)} = vel^{IWD}_{(t)} + \Delta vel^{IWD} \tag{6.2}$$

The amount of soil removed from the bed of edge (i,j) is inversely and non-linearly proportional to the time needed for the IWD to move from point i to point j and can be calculated by using Equation (6.3)

$$\Delta soil(i,j) = \frac{a_s}{b_s + c_s.time^2(i,j;vel^{IWD})} \tag{6.3}$$

where a_s, b_s and c_s are pre-defined positive parameters. a_s and c_s define the relationship between the amount of soil and the period of time IWD takes to move through the edge (i,j), and b_s is a small number used to avoid the singularity problem. Meanwhile, the duration of time is calculated by the simple laws of physics for linear motion. Thus, the time spent by the IWD to move from point i to point j is inversely proportional to the current velocity of the IWD.

$$time(i, j; vel^{IWD}) = \frac{HUD(i, j)}{\max(\epsilon_v; vel^{IWD})} \tag{6.4}$$

where a local heuristic function $HUD(i, j)$ has to be defined for a given problem to measure the undesirability of an IWD to move from point i to point j, ϵ_v is the threshold of velocity to avoid the negative value of vel^{IWD}. Equation (6.3) and (6.4) represent the assumption that the water drop which moves faster or spends less time to pass from point i to point j can gather more soil than the one which has a slower velocity.

The following formulae are used to calculate the updated soil of the edge and the soil load of the IWD respectively.

$$soil(i, j)_{(t+1)} = (1 - \rho_n).soil(i, j)_{(t)} - \rho_n.\Delta soil(i, j) \tag{6.5}$$

$$soilLoad^{IWD}_{(t+1)} = soilLoad^{IWD}_{(t)} + \Delta soil(i, j) \tag{6.6}$$

where ρ_n is the local soil updating parameter, which is chosen from $[0, 1]$.

To present the behaviour of an IWD that it prefers the easier edge or the edge with less soil on their beds, the edge selection of an IWD is based on the probability, $p(i, j; IWD)$, defined as follows which is inversely proportional to the amount of soil on the available edges.

$$p(i, j; IWD) = \frac{f(soil(i, j))}{\sum_{k \notin vc(IWD)} f\big(soil(i, k)\big)} \tag{6.7}$$

where $f\big(soil(i, k)\big) = \frac{1}{\epsilon_s + g\big(soil(i,j)\big)}$. The constant ϵ_s is a small positive number to prevent singularity. The set $vc(IWD)$ denotes the group of nodes that the IWD should not visit to satisfy the constraints of the problem. The function $g\big(soil(i, j)\big)$

is used to shift the $soil(i,j)$ of the edge connecting point i and point j towards a positive value and is described below

$$g\big(soil(i,j)\big) = \begin{cases} soil(i,j) & \text{if } \min\limits_{l \notin vc(IWD)} \big(soil(i,l)\big) \geq 0 \\ soil(i,j) - \min\limits_{l \notin vc(IWD)} \big(soil(i,l)\big) & \text{otherwise} \end{cases} \tag{6.8}$$

A uniform random distribution is used to generate a random number which can be compared to this probability in order to decide which is the next location where the IWD will move to. The IWDs work together to find to the optimal solution. For a given problem, an objective or quality function is needed to evaluate the fitness value of the solutions. The function $q(.)$ is denoted as the quality function and T^{IWD} is a solution founded by the IWD, the iteration best solution T^{IB} is identified by

$$T^{IB} = \arg\max_{\forall IWD} q(T^{IWD}) \tag{6.9}$$

In order to increase the opportunities of finding the global optimum, the amount of soil on the edges of the iteration best solution T^{IB} is updated according to the goodness of the solution. The following formula can be used to update the $soil(i,j)$ belonging to the iteration best solution T^{IB}

$$soil(i,j) = (1+\rho_{IWD}).soil(i,j) - \rho_{IWD}.\frac{1}{(N_{IB}-1)}.soilLoad_{IB}^{IWD}$$

$$\forall (i,j) \in T^{IB} \tag{6.10}$$

where $soilLoad_{IB}^{IWD}$ represents the soil of the iteration best IWD when it reaches to the destination, N_{IB} is the number of nodes in the solution T^{IB}, and ρ_{IWD} is the global soil updating parameter which is chosen from $[0,1]$.

6.4 Optimal Data Aggregation Tree Formation of WSNs using Intelligent Water Drops Algorithm

6.4.1 Constructing aggregation tree with IWD algorithm

The original IWD algorithm is used to solve the classical TSP in order to find the shortest path that connects all the cities in the problem exactly once, returning to the first city [27]. In the problem of constructing the optimal data aggregation tree for wireless sensor networks, the IWD algorithm can be adopted in a similar way. However, different from TSP problem where an IWD travels via all the nodes to establish a circle, several IWDs are originated from data sources simultaneously to search for the paths to reach the BS. These paths join and form an aggregation tree which is a near optimal solution of the problem described in Section 6.2. The basic idea is that each IWD either tries to either find the shortest path to reach the destination, or find the nearest aggregation nodes and terminates. Each sensor node node maintains a table in its memory storage that contains the information of its neighbors' identities and the amount of soil of soil assigned to each edge connecting it to the neighbor nodes. Initially, the same amount of soil is assigned to every edge, which connects two nodes in their mutual neighborhood. The algorithm updates the soil of each edge along the route to the BS whenever an IWD traverses. The better route, including a smaller number of hops from source to the destination, has less amount of soil left and thus attracts more IWDs in the next round of search.

The format of packet carried by IWDs is shown in Table 6.1 to provide the information to update the soil of the routes visited. In this packet, the field Pkt_Type indicates the type of the packet which is a *control packet* or a *data packet*. The field SN is the selected node which receives the packet. S_ID is the identity of the source node generating this packet. P_ID is the identity of the previous node. HCtoS contains the hop count from the current node, where the IWD is locating, to the source node. The last two fields represent the parameters of the IWDs: IWD's velocity, IWD_vel and soil load, IWD_soilLoad. The initial value of IWD's velocity and soil load is set by the constants $InitVel$ and 0 respectively at the source nodes. This value changes when IWD packet travels through the network.

Table 6.1: Message format of an IWD packet.

Pkt_Type	SN	S_ID	P_ID	HCtoS	IWD_vel	IWD_soilLoad

Initially, the BS floods its identity throughout the network. Each node in the network receiving this packet broadcasts to its neighborhood. The hop count to the BS of a node is updated if it receives the packet from a neighbor which has a smaller hop count to the BS than its current hop count subtracting one.

The IWD algorithm proposed for WSNs illustrated in Fig. 6.2 is described in the following steps:

Step 1: Each source node initializes its IWD data packet. The initial value of this amount of soil is $soil(i, j) = InitSoil$.

Step 2: Each IWD data packet located at node i selects the next hop j from its neighbor table by generating a random number and comparing with the probability calculated with the Equation (6.7). In order to eliminate the selection of nodes, which is further away from the BS, an additional term, $h_d(i, j)$, is included into

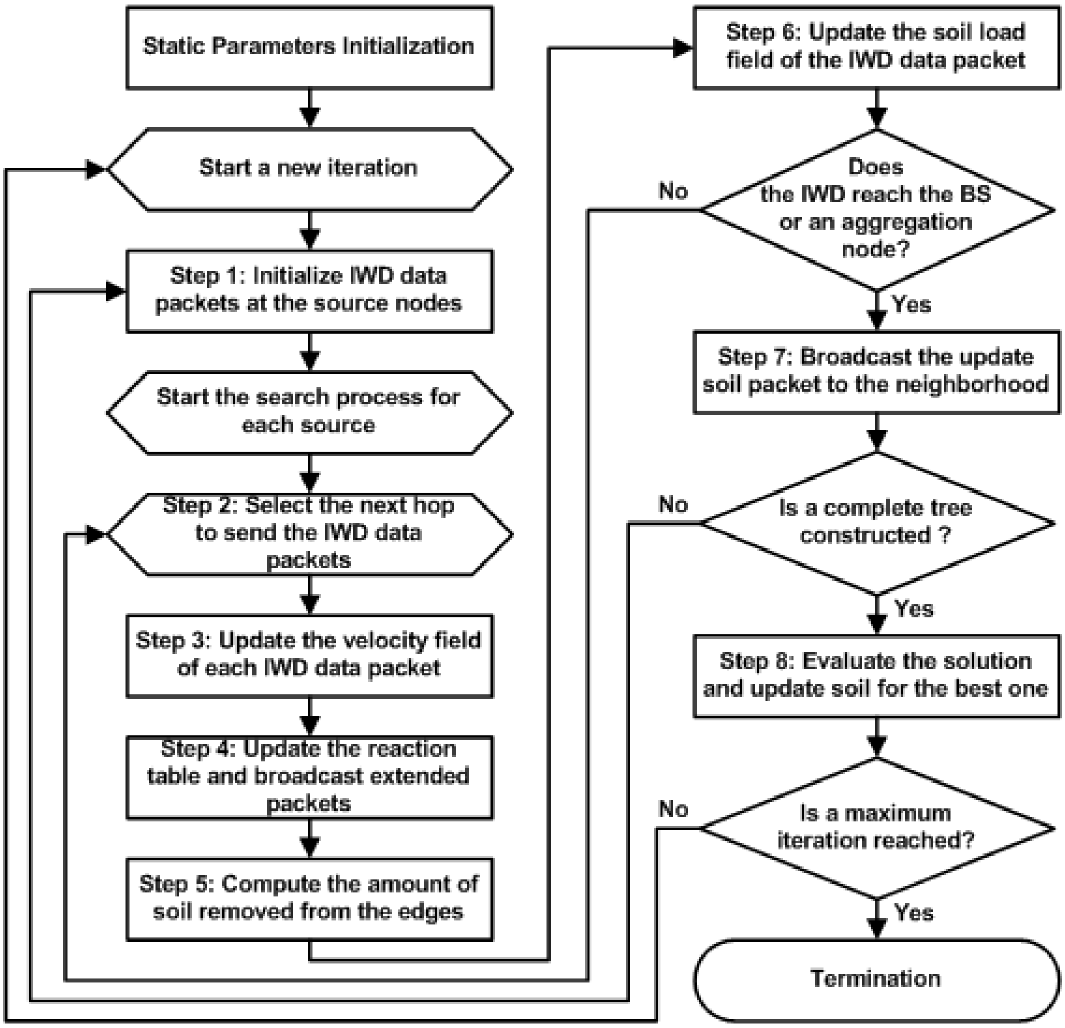


Figure 6.2: The flowchart of the IWD algorithm for constructing data aggregation tree in WSNs.

$f\big(soil(i,j)\big)$ as follows

$$f\big(soil(i,j)\big) = \frac{1}{\epsilon_s + g\big(soil(i,j)\big)} \cdot h_d^{-\beta}(i,j) \tag{6.11}$$

where $h_d(i,j)$ is the hop count from node j to the BS adding one and β is a parameter which determines the relative influences of $\eta(i,j)$.

Step 3: Node j receiving IWD data packet updates the velocity of the IWD by using Equation (6.2).

Step 4: When node j receives an IWD packet, it builds a reaction table which consists of five parts:

- S_ID: source node id

- P_ID: previous node id

- HCtoS: hop count from node j to the source node

The S_ID and P_ID fields of the reaction table are the same values as those in the IWD packet. This information is used to extend the search region, which follows a mechanism proposed in [34] in order to increase the probability of finding the intersection of the routing paths. HCtoS or the table is obtained by adding one to HCtoS value in IWD packet. However, if the S_ID already exists in the record of node j, the reaction table only update if the new HCtoS is smaller than the old one.

After updating the reaction table with the new information of the IWD data packet, the node broadcasts an extended packet in one hop communication to its neighbors. This packet has the same format as IWD packet except that the data field and the fields of IWD properties are empty. The neighbor nodes in turn update their reaction table accordingly.

Step 5: The amount of soil removed from the edge (i, j) is computed by using Equation (6.3). The undesirability of an IWD to move from node i to node j,

$HUD(i, j)$, in this equation is defined as follows

$$HUD(i, j) = \sum_{k \in R_j} hs_{kj} + hd_j \tag{6.12}$$

where R_j is the set of source nodes that are included in the reaction table of node j, hs_{kj} is the hop count from node j to source node k and hd_j is the hop count from node j to the destination or the BS. Hence, the time spent by IWD to travel from node i to node j is directly proportional to the total hop count to all the source nodes and to the BS. If this total hop count is small, the time to travel is also small that results in a larger amount of soil being removed from the edge (i, j).

Step 6: The value of IWD_soilLoad in the IWD data packet is updated by using Equation (6.6). The soil of the edge is calculated by using the following equation modified from Equation (6.5).

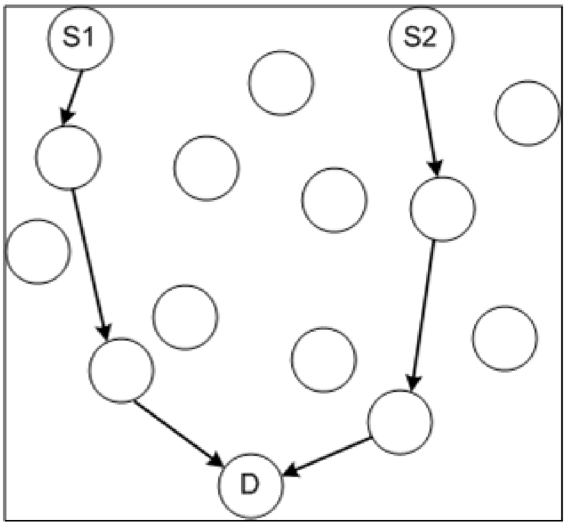
$$soil(i, j) = (1 - \rho_n).soil(i, j) - \rho_n.(1 + hd_i - hd_j).\Delta soil(i, j) \tag{6.13}$$

The term $(1 + hd_i - hd_j)$ is added in order to eliminate the edge which lets IWD move further away from the BS. If the hop count from node i to the BS is larger than that of node j, node j is closer to the BS. As a result, more amount of soil is removed from the edge (i, j). Otherwise, no change of the amount of soil is applied.

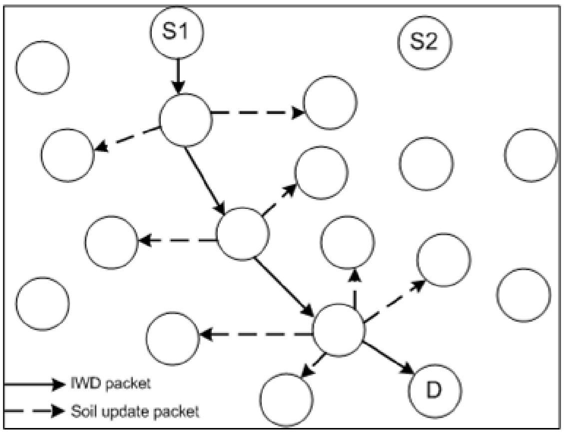
Step 7: Once the IWD data packet reaches the BS or an aggregation node, it terminates the search process. Otherwise it repeats from Step 2 to select the next hop. At the aggregation node v, the amount of soil in each edge (u, v), where u is a neighbor of v, is updated as follows. The aggregation node v broadcasts an update soil packet to all the neighbor nodes except the parent node which is the previous one sending data to v. This packet contains the total of hop count from all the sources to v and from v to the BS. Each of these neighbor nodes in turn will create a new IWD which has the velocity of InitVel and soil load of 0. The IWD velocity and soil load is calculated by using Equation (6.2) and (6.6). Then they are applied into Equation (6.5) to update the $soil(u, v)$.

Step 8: The network operation repeats from step 1 to step 7 in order to collect data for a number of iterations. For each of iterations, after the data aggregation tree, rooted at the BS and containing all the source nodes, is constructed by IWD algorithm, the BS computes the total hop count of the data aggregation tree. If the total hop count is smaller than the best value evaluated so far, soil of all edges in the aggregation tree is updated one more time by using Equation (6.10).

This basic IWD algorithm as explained above is able to construct a data aggregation tree with the minimum number of hop count. The algorithm increases the probability of finding the best aggregation node by updating the amount of soil for the neighbor of the aggregation node once it is found. Thus, in the next round, there is a higher chance that one IWD moves through this aggregation node when this IWD reaches its neighborhood. However, there are cases when an IWD cannot find any nodes that are visited by other IWDs, thus, the route established by this IWD does not have any common point in the middle with others. As illustrated in Fig. 6.3a, the routes from the source node s_1 and the source node s_2 to the BS, D only join together at D. The attraction of joining together for other IWDs is only performed at the BS that is far away from these sources. Therefore, the probability of finding an aggregation node which is near to the sources is less. Higher energy is required for transmitting data from sources to the BS, due to the fewer chances of constructing an optimal aggregation tree.



(a) Routing data without applying improvement.



(b) Routing data with applying improvement of updating soil throughout the route.

Figure 6.3: The process of routing data from source nodes to the destination or BS with and without improvement.

6.4.2 Improvement of the IWD algorithm for searching the aggregation nodes

In order to increase the probability of reaching an aggregation node for other IWDs, an IWD should update the amount of soil for the neighbors of all the nodes along the route it creates instead of only updating the amount of soil for the neighbors of the aggregation nodes as shown in Fig. 6.3b. Whenever a node v receives an

IWD data packet, it broadcasts an update soil packet to its neighborhood. Each neighbor node u uses the information in this packet to update the $soil(u, v)$ in the same way as presented in Step 7. of the algorithm in Section. 6.4.1. This approach is expected to enhance the chance of bringing the aggregation nodes closer to the sources after fewer runs so that data can be aggregated and thus higher energy efficiency is achieved. For example, an IWD packet containing data originates from source nodes $S1$ and needs to find a route to the BS b. When it moves from a relay node i to another node j, in order to update the amount of soil in node j's neighborhood, it broadcasts an update soil packet. The value of $soil(i, j)$ is updated as in Equation (6.5). For the other neighbor node k of j, if its hop count to the BS is less than that of node j, the amount of soil of the edge (k, j) will not be updated. Otherwise, node k generates a new IWD' with soil load is zero and velocity is set to InitVel. The velocity of this IWD' is later updated as below

$$vel^{IWD'} = InitVel + \frac{a_v}{b_v + c_v.soil^2(k, j)} \tag{6.14}$$

Thus the amount of soil will be removed from the edge (k, j) is

$$\Delta soil(k, j) = \frac{a_s}{b_s + c_s.\left(\frac{HUD(k,j)}{vel^{IWD'}}\right)} \tag{6.15}$$

And the soil of the edge (k, j) is updated by the following formula

$$soil(k, j) = (1 - \rho_n).soil(k, j) - \rho_n.(1 + hd_k - hd_j).\Delta soil(k, j) \tag{6.16}$$

By repeating this process for all of the nodes along the route that the IWD packet from the source node $S1$ traverses to reach the BS b, the probability of being chosen as an aggregation node for all of these nodes are increased proportionally to their hop count to the BS and all of these nodes are potential to become an aggregation node. IWD packets originated from other source nodes tend to select the nodes of the route from $S1$ to b as the next hop to pass the data when they move in the neighborhood of this route. And because the number of potential aggregation nodes is also enlarged, there is high chance for an IWD to encounter a node, which has been visited by another IWD.

6.5 Computational Experimental Results and Discussions

In this section, the simulation results of the basic and improved-IWD algorithms are presented and compared with the performance of ACO implemented in [34]. The simulation parameters are given in the Table 6.2.

Table 6.2: Simulation parameters.

Parameters	Values	Description
N	300	Number of sensor nodes
S	$5 - 30$	Number of source nodes
R	$10 - 12\,m$	Communication range
E_{elec}	$50\,nJ/bit$	Energy consumption of transceiver electronics
E_{fs}	$0.01\,nJ/bit/m^2$	Energy consumption of transmitter amplifier
ctrlPkt	8 byte	Length of *control packet*
dataPkt	250 byte	Length of *data packet*
E_{init}	$0.5J$	Initial energy of each node

The initial settings of the parameters used in the IWD and improved-IWD algorithms are shown in Table 6.3. These values are chosen as the same ones used in [28].

During the computational experiments, a WSN with a number of N sensor nodes are randomly deployed in a square area $100\,m \times 100\,m$. All the sensor nodes

Table 6.3: IWD algorithm parameters.

Parameters	Values	Description
$InitSoil$	10000	Initial soil of each edge
$InitVel$	200	Initial velocity of an IWD
a_v, b_v, c_v	1, 0.01, 1	Velocity updating parameters
a_s, b_s, c_s	1, 0.01, 1	Soil updating parameters
ϵ_s	0.01	-
ϵ_v	0.0001	Positive velocity threshold
ρ_n	0.9	Local soil updating parameter
ρ_{IWD}	0.9	Global soil updating parameter
β	20	-

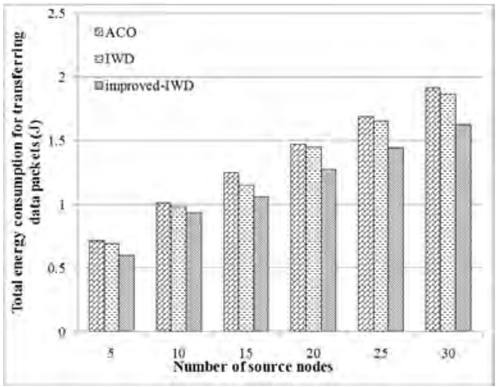
have the same communication range, R; the source nodes collect data periodically and send towards a single destination node, which is the BS, located at coordinates $(10\,m, 10\,m)$ of the deployment field as shown in Fig. 6.1a.

Case 1: Comparative study of total energy consumption

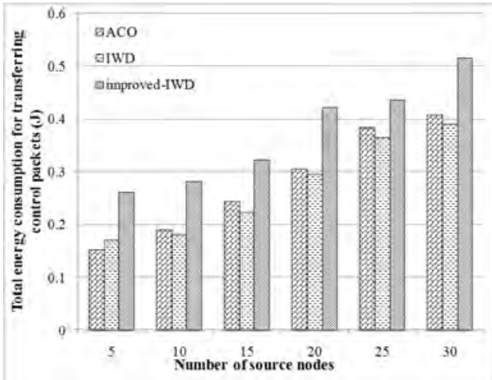
The first computational experiment compares the total energy consumption for transferring information within the WSN after the same number of runs for different algorithms. The test results shown in Fig. 6.4a, 6.4b and 6.4c are obtained with 100 runs for a WSN of $N = 300$ sensor nodes. Comparisons are made for the networks with different number of source nodes.

Fig. 6.4a shows the energy consumption of the WSNs for transmitting *data packets* when using ACO, IWD and improved-IWD algorithms. It is observed that the WSN using IWD consumes less energy than the network with ACO, and improved-IWD is able to further enhance energy conservation when compared with these two. This is due to the fact that a the fact that a smaller number of hops are involved in the data transfer with IWD and improved-IWD when compared with

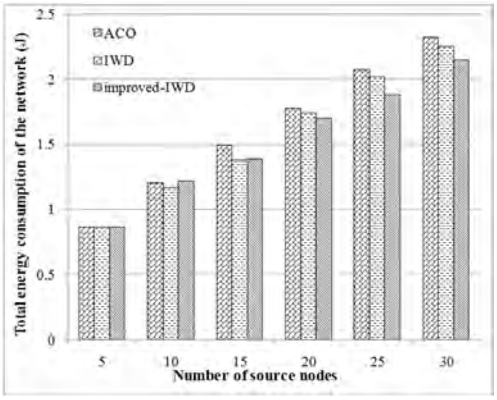
ACO. For example, when the number of source nodes chosen is 20, the amount of energy consumed for transmitting *data packets* by using IWD and improved-IWD algorithms are 1.83% and 18.21% respectively less than that of the WSN using ACO.



(a) Total energy consumption for transferring *data packets*.



(b) Total energy consumption for transferring *control packets*.



(c) Overall total energy consumption of the network.

Figure 6.4: Total energy consumption for transferring information within the network.

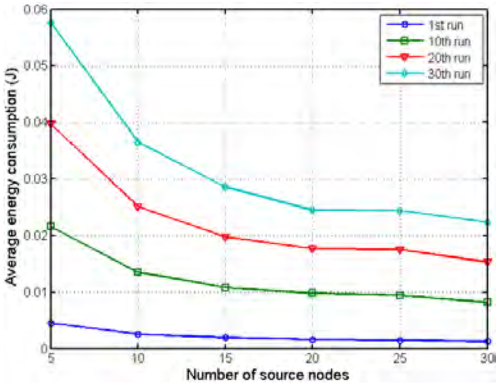
The total energy consumption of transmitting and receiving *control packets* is shown in Fig. 6.4b. Compared with ACO, the amount of energy used for *control packets* in WSN with IWD algorithm is slightly smaller. The reason is that the data aggregation tree obtained by IWD converges to a solution which contains a fewer number of relay nodes and thus, a smaller number of *control packets* are required. Fig. 6.4b also shows that the network with the improved algorithm of IWD consumes the most energy for *control packets*. This consequence is due to the fact that there are more *control packets* needed to update the soil of the edge in the neighborhood of the route founded by IWDs. Sensor nodes in the neighborhood have to consume more energy to receive these update soil packets. In addition, when the number of source nodes is small, that means there are fewer data routes connected from sources to destination, there is not much difference among the results obtained by all the algorithms. In this case, the ratio between the cost of constructing the tree and the energy consumption saved in data transfer is high. When the number of source nodes increases, the amount of data transferred within the network is more. Thus, by applying the IWD and improved-IWD, more amount of energy for transmitting *data packets* is saved, since the formation data aggregation tree is enhanced and more data is aggregated. This energy saving surpasses the cost of optimizing the aggregation tree, i. e. the energy consumed to transmit *control packets*. Fig. 6.4c shows that the total energy consumption of the network when using IWD algorithm is less than that of ACO and the result achieved by the improved algorithm is the least, especially with the larger number of source nodes. For example, when the number of source nodes is 20, the percentage of the total amount of energy saving by using IWD and improved-IWD algorithms are 1.99% and 5.38% when compared with ACO, respectively.

Case 2: Comparative study of average energy consumption

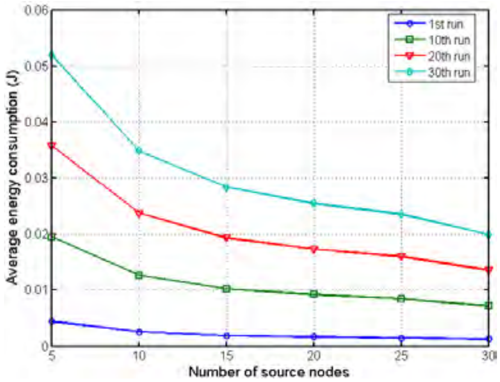
The next experiment investigates the average amount of energy consumption for each source nodes to transfer data to the BS. The measurement is carried out for the network using ACO, IWD and the improved algorithm of IWD with the network of $N = 300$ and the communication range $R = 12\,m$. The results of ACO, IWD and improved-IWD are shown in Fig. Fig. 6.5a, 6.5b and 6.5c respectively. After the first run, the average energy consumption is almost constant for the different number of source nodes. This happens due to the fact that the data aggregation tree is randomly constructed, thus the average energy spent for each source node to transfer data is equal to networks with the various number of source nodes. After 10th, 20th and 30th runs of the network operation, the aggregation tree is gradually optimized. With the higher number of source nodes, there is more amounts of data aggregated in the optimal tree. Hence, the total amount of data transferred within the network is decreased that result in the reduction of average energy consumption per source node. It can be seen that with the number of source nodes of 30, the average energy consumption per source node with improved-IWD is the least as compared to that with basic IWD and ACO algorithms.

s *Case 3: Comparative study of the network lifetime*

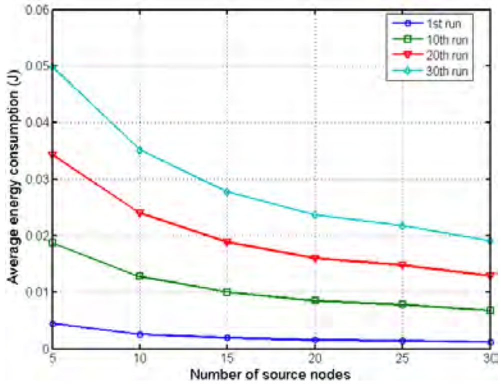
The network lifetime is studied in this computational experiment. The network lifetime is defined as the maximum number of runs of network operation until the first node runs out of energy. Fig. 6.6 shows the maximum number of runs when different algorithms are used for the same network deployment with $N = 300$ and $R = 12\,m$. The network lifetime is slightly better with IWD and significantly enhanced with its improved version when compared to ACO. For example, with the number of source nodes of 30, the lifetime of the networks with ACO, IWD and improved-IWD algorithms are 831, 880 and 952 respectively. This is due to the result of more aggregation nodes found, which are nearer to the sources, and the less energy consumed by the BS's neighboring nodes for receiving only the aggregated



(a) ACO alogrithm.



(b) IWD algorithm.



(c) improved-IWD algorithm.

Figure 6.5: Average energy consumption of the network when using different algorithms.

data.

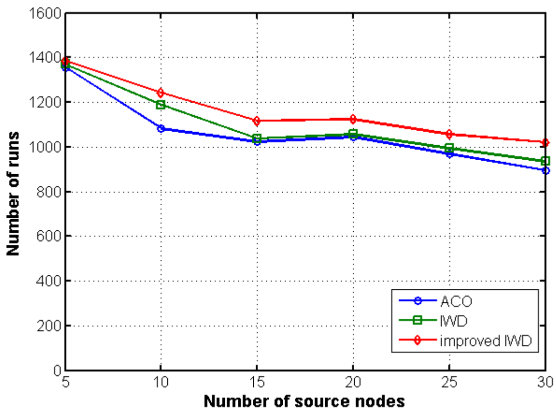
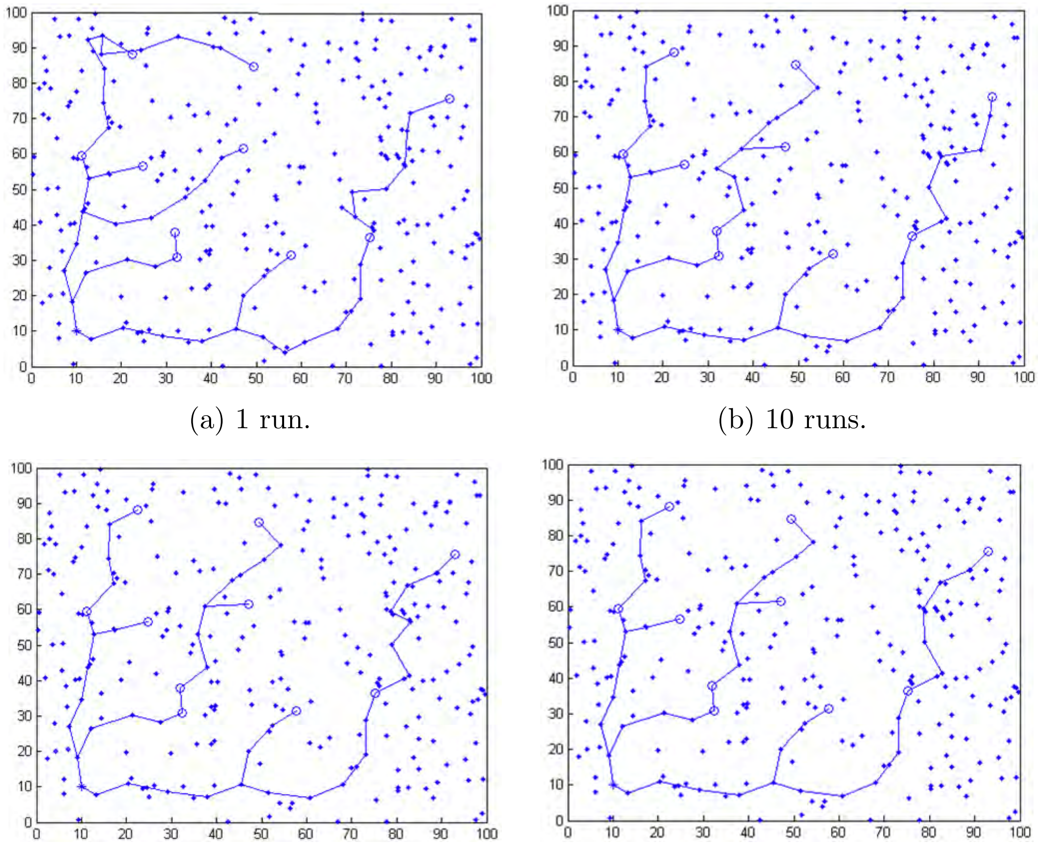


Figure 6.6: Comparison of network lifetime with different algorithms.

Case 4: Investigation of the aggregation tree construction



(a) 1 run. (b) 10 runs.

(c) 20 runs. (d) 30 runs.

Figure 6.7: The formation of the data aggregation tree after different number of runs.

Fig. 6.7a, 6.7b, 6.7c and 6.7d illustrate the data aggregation tree constructed after 1, 10, 20 and 30 rounds of operation, respectively. In this simulation, $N = 300$ sensor nodes are used, number of source nodes, S is 10 and communication range of each sensor node, R is $12\,m$. After a number of rounds, the optimal data aggregation tree is constructed. Source nodes are able to transmit data towards the base station through the best route; data is aggregated and fused at the optimum aggregation nodes.

The simulation is run for 30 times with the same network deployment to investigate the performance of the algorithms. The best total number of edges, hc_b, and the average total number of edges in the data aggregation tree, hc_{ave}, found by

Table 6.4: The total hop count of data aggregation tree

Number of source nodes	**ACO**		**IWD**		**improved-IWD**	
	hc_b	hc_{ave}	hc_b	hc_{ave}	hc_b	hc_{ave}
5	28	31.7	28	30.3	24	25.3
10	45	52.7	44	51.6	42	45.5
15	55	62	52	61.7	48	55.1
20	65	70	65	69.7	57	60.4
25	73	75.3	73	74.3	62	64.3
30	84	85.7	83	84.3	72	73.8

using the proposed algorithms and ACO are shown in Table 6.4. With the different number of source nodes, the performance of the basic IWD algorithm is slightly better than ACO algorithm, the data aggregation trees with fewer number of edges are constructed with IWD. Moreover, the results obtained by the improved-IWD outperform both IWD and ACO algorithms. When the number of source nodes is 30, the average total number of edges with improved-IWD is 12.45% and 13.88% less than IWD and ACO respectively. That best value with improved-IWD is 13.25% less than IWD and it is 14.29% smaller when compared to ACO.

Case 5: Complexity analysis

In order to analyse the complexity of the algorithms, the computational time taken by each algorithm is also investigated. The WSNs of 300 nodes and the different number of source nodes is taken into account. Simulations are carried out for 500 runs for each number of source nodes. The average time taken for each run of the simulation by each algorithm is shown in the Table 6.5.

It is evident in the results that the computational time is almost same for IWD and ACO algorithms, and it takes a bit longer time for improved-IWD during

Table 6.5: The average computational time, in *(second)*, per iteration taken by different algorithms

Number of source nodes	5	10	15	20	25	30
ACO algorithm (in sec)	0.0153	0.0205	0.0249	0.0327	0.0330	0.0350
IWD algorithm (in sec)	0.0153	0.0209	0.0252	0.0328	0.0332	0.0351
improved-IWD -algorithm (in sec)	0.0157	0.0219	0.0265	0.0340	0.0347	0.0360

the simulation. With the number of source nodes of 30, the computation time with improved-IWD and IWD are 2.85% and 0.29%. This is due to the fact that there are more IWDs required to update the soil of the edges connected the neighboring nodes with the main routes relaying data. However, the difference is insignificant; the improved-IWD does not impose much extra computational burden and the computational time is comparable with that of IWD and ACO algorithms.

6.6 Conclusion

In this chapter, the intelligent water drop algorithm is investigated to solve the problem of optimal data aggregation trees for wireless sensor networks. The computational experiment results show that improved-IWD and IWD algorithms can achieve good results comparable with ACO algorithm in terms of energy efficiency. By using IWD algorithm, the number of edges of the of the aggregation tree is reduced, and thus the overall network consumed less energy. Further improvement of the original IWD original IWD algorithm is carried out by updating the soil of the sensor nodes' neighborhood when the routes are established in order to increase the chance of selecting optimal aggregation nodes. Although the cost of constructing the optimal trees, i.e. energy consumption for transferring *control packets*, is increased, the total energy consumption of the network with improved-IWD algorithm is still less than that with ACO and IWD algorithms. In addition, in all investigated scenarios, better aggregation trees of the WSNs are obtained with the improved-IWD algorithm, meanwhile insignificant complexity is added. The achievement of energy conservation results in the longer lifetime of the networks with IWD and further enhancement of the lifetime with improved-IWD algorithm.

Chapter 7

Conclusions and Future Works

This chapter concludes the thesis. It briefly restates the motivation of the thesis work, the identified problem areas and the various findings in each problem area. Finally, it shows the the direction of future research in this regard.

7.1 Conclusions

This thesis covers work carried out on development of energy efficient routing protocols for Wireless Sensor Networks. Due to the sensor nodes' small size and ability of wireless data transfer, the WSNs can be employed in a vast area of applications. However, the WSNs have very constraint resources, i.e. small memory, limited computation capability and finite energy sources. Therefore, it is challenging to design and operate the WSNs efficiently, especially in terms of energy. The IEEE 802.15.4 standard has defined a standard for PHY and MAC layers for WSNs, but left the NWK layer, i.e. the routing layer for developer. Although, there have been a number of routing protocols proposed in the literature, few protocols have been implemented in practice. Besides, further enhancement of these protocols is needed to be carried out. Chapter 1 provides an overview of the WSNs and the problem definition that would be solved in this thesis. A thorough review of main challenges and technical aspects of WSNs are presented in Chapter 2. The two fundamental types of routing protocols are focused: *cluster-based routing protocols* and *tree-based routing protocols*. The *cluster-based routing protocols* can achieve better energy savings, since the amount of data transferred is significantly reduced and the cluster members only need to turn on radio for a very short period of time. However, this type of protocol is suitable when communication range of the sensor nodes is able to cover the deployment field and transmit control packets directly to the BS during the network setup phase. Meanwhile, the *tree-based routing protocols* enable multihop communication in the network, thus the communication range can be extended, eventhough, it is required more energy for routing data and is not able to synchronize the on and off time of radio operation to achieve energy savings. The motivation of this thesis was to develop high energy efficient routing protocols for WSNs.

It is essential to have a good understanding of the WSNs energy consumption and the characteristics of the sensor nodes before developing the protocols. Chapter 3 described an energy model popularly employed to investigate energy usages of the network in simulation. A simulation test is performed to study the energy efficiency of cluster head rotation scheme in cluster-based WSNs. A sensor node is also designed to apply in health-care medical systems with simple direct communication. The sensor node is equipped with a renewable energy source that harvests energy from human warmth in order to power the node for infinite period of time. For complicated communication requirement that the sensor node can talk to each other and form a larger network, a more complex hardware platform, IRIS, is adopted and studied. The platform is supported by an embedded networking operating system, called TinyOS, that is designed to be lightweight and low power operation. The TinyOS application and systems are written in nesC language, which is a C dialect with features to reduce RAM and code size and enables significant optimization. The analysis of IRIS node power consumption for communication is performed. The influence of power transmission level adjustment on the energy consumption and communication efficiency is investigated. This information can be used later for configuring the network operation. Further study of cluster head rotation is carried out in practice on the IRIS platform with an efficient mechanism of intra-cluster power management. The power management strategy proposed in Chapter 3 utilizes battery aware selection of cluster head and a loose synchronization scheme among the normal nodes with the cluster head. Experimental results are provided to prove the efficiency of the proposed strategy when compared with conventional selection schemes of the cluster head within one cluster.

Chapter 4 proposed a complete cluster-based routing protocol for WSNs using the FCM clustering algorithm. Since such typical probabilistic cluster-based protocols like LEACH construct unequally distributed cluster formation, the traffic load and energy consumption of the CHs are very high that results in fast depletion of the CHs' energy sources and shortens the network lifetime. The FCM clustering algorithm allocates sensor nodes into clusters in a fuzzy manner in order to minimized the intra-cluster mean distance of the network. It has been proved in the simulation results that uniform cluster construction of the network is achieved. The CHs selection is made by choosing the node with the highest residual energy in each cluster. As shown in the simulation results, the average energy consumption of the network is reduced and network lifetime is extended by using FCM when comparing with other conventional protocols. Additionally, a framework for designing centralized cluster-based protocols is proposed. Based on this framework, practical FCMCP for WSNs is also developed on the top of the ActiveMessageC layer in TinyOS. The experiment results show a successful operation of a real-time FCMCP network. Furthermore, long lifespan of the network is also achieved when compared with LEACH.

In order to achieve optimal cluster formation and cluster head selection during the setup phase in the cluster-based WSNs, a cluster-based protocol using a nature-inspired optimization algorithm is proposed in Chapter 5. An objective function is formulated to represent the optimization goal that is to obtain an uniform distribution of sensor node into clusters, meanwhile the cluster heads can be chosen optimally in terms of energy proficiency. The novel HSA is adopted to solve this problem efficiently. The HSA mimics the process of improvising music in which the musician searches for harmony and continues to polish the pitches to obtain a better harmony. Applying into the problem of forming the optimal clusters and selecting the best CHs, the HSA attempts to find the solution that can minimize the formulated objective function. The consequence is that the intra cluster mean distance is minimized and the CHs are chosen with the compromise between uniform cluster formation and energy awareness. This strategy guarantees high energy efficiency during the network operation and assists to avoid fast depletion of the nodes that have been chosen as CHs. The comparison of the HSA performance have been made with other well-known evolutionary algorithms like GA and PSO. It has been shown that the HSA is able to achieve better fitness value with faster convergence when solving the formulated problem. Simulation is also carried out to investigate the application of the proposed protocol using HSA when it is employed to operate the WSNs. An extended network lifetime is achieved when compared with other conventional clustering protocols. Validation on hardware test-bed for HSACP is also performed. The framework proposed in Chapter 4 is utilized to develop the real-time HSACP based clustering protocol. The HSA is implemented in Java programming language and executed on a computer connected to a BS node. The study of the algorithm execution shows that computational time is fast enough to setup the network in practice, meanwhile the optimal solution can be obtained. It can be seen from the experimental results the HSA based clustering protocol successfully operates on a 30-node WSN. The duration of the setup phase is comparable with other protocols such as LEACH like protocol and FCMCP. On the other hand, the HSACP also enables the WSN to run for longer period of time in a real-life application like indoor environment monitoring.

Although the cluster-based protocols are able to operate the network in a high energy efficient manner, they have some drawbacks. One of the main drawbacks is the limitation of the communication range, it is required the sensor nodes have the ability of direct communication with the BS during the network setup phase. To overcome this issue, multihop communication range can be employed. Additionally, the data aggregation technique is still necessary to reduce the number of data packets transferred within the network that means lower energy consumption is required. Therefore, a tree-based routing protocol is proposed in Chapter 6. The objective of this protocol is to establish an optimal data aggregation tree connecting all the sources that generate data with a BS. This objective is satisfied by applying a nature-inspired optimization algorithm call IWD. The IWD algorithm imitates the flows of the water drops in the river that tends to find the easiest routes with the least obstacle to moves. The basic version of IWD algorithm is applied to solve the problem of constructing the best data aggregation tree and selecting the optimal aggregation node. Further improvement is also achieved with a modification of the protocol in order to find the aggregation node in a faster way, and thus more energy savings is obtained. The simulation results show that the IWD and its improved version are able to enhance the efficiency of the energy usages by constructing better data aggregation tree when compared to a similar protocol using a well-known optimization method such as ACO. Meanwhile, the energy consumption for establishing the tree is increased, the amount of energy that can be saved during the data transmission phase is significantly reduced by applying IWD and improved-IWD when compared with ACO. Therefore, total energy consumption of the networks that use IWD and improved-IWD is still less. Besides, the network lifetime is also enhanced.

In all, the main objectives as laid out in Chapter 1 of this thesis have been achieved. The findings of this work have been published in international technical conferences and journals for the benefit of future researchers and users of energy efficient protocols for WSNs with the support of the optimization methods such as FCM, HSA and IWD. A list of the publications from this thesis work is provided.

7.2 Future works

Optimizing the operation of cluster-based routing protocols with the consideration of Quality of Services There are many improvements that can be done to the existing protocols. The optimization of the network construction can be enhanced with the consideration of different objectives so that not only the energy consumption is minimized but the quality of service is guaranteed. Other aspects of the WSNs operation quality such as data throughput, fairness and latency can be incorporated into the optimization problem.

Localization service for self identifying position of the sensor nodes The position of the sensor nodes is necessary for optimizing the network, especially in cluster-based protocols. However, there is difficulty in identifying the exact place where the sensor node locates. Since the GPS module consumes significant amount of energy and is not available for indoor applications, GPS seems not to be suitable for the WSNs. RSSI is potential in the case of WSNs, most of the transceivers used in wireless sensor nodes are able to measure the strength of the received signal. It is also studied in Chapter 3 that the RSSI can be used to determine the relative distance from the sender node to the receiver node. However, the accuracy of this technique is not very high and gets effected by the transmitting environment. For example, in indoor environment, the attenuation of the signal strength varies due to obstacles or reflection from the walls. Therefore, it is necessary to develop algorithms to deal with the noise and process the RSSI values properly so that the precise location of the sensor node can be obtained in an energy efficient way.

Incorporation of renewable energy sources in WSNs The renewable energy sources can be considered as infinite source that theoretically is able to power the sensor nodes forever. However, in most of the cases, the energy harvested from the ambient environment is fluctuating and intermittent. Therefore, the energy of the sensor node still needs to be used wisely. It is required to manage the harvesting process in an intelligent way and cooperate it tightly with the operation of the systems.

Publications

Published

Journals

1. D. C. Hoang, R. Kumar, S. K. Panda, ”Optimal Data Aggregation Tree for Wireless Sensor Network using Intelligent Water Drop Algorithm”, *IET Journal of Wireless Sensor Systems*, 2012.

Conferences

1. D.C. Hoang, Y.K. Tan and S.K. Panda, ”Thermal Energy Harvesting From Human Warmth For Wireless Body Area Network In Medical Healthcare System”, *The Eighth International Conference on Power Electronics and Drive Systems (PEDS09)*, Taiwan, 2009.

2. D. C. Hoang, R. Kumar, S. K. Panda, ”Fuzzy C-Means Clustering Protocol for Wireless Sensor Networks”, *IEEE International Symposium on Industrial Electronics (ISIE-2010)*, Bari, Italy, 2010.

3. D. C. Hoang, P. Yadav, R. Kumar, S. K. Panda, ”A Robust Harmony Search

Algorithm based Clustering Protocol for Wireless Sensor Networks", *ICC'10 Workshop E2Nets*, Cape Town, South Africa, 2010.

4. D. C. Hoang, K. T. Ching, R. Kumar, S. K. Panda, "Intra-Cluster Power Management for Wireless Sensor Networks", *IEEE International Symposium on Industrial Electronics (ISIE-2012)*, Hangzhou, China, 2012.

5. D. C. Hoang, S. K. Panda, "Real-time Power Configuration for Energy Conservation in Wireless Sensor Networks", *IEEE International Conference on Communication Systems (ICCS-2012)*, Singapore, 2012.

Accepted

Journals

1. D. C. Hoang, R. Kumar, S. K. Panda, "Realization of a Cluster-based Protocol using Fuzzy C-Means Algorithm for Wireless Sensor Networks", *IET Journal of Wireless Sensor Systems*, 2013.

2. D. C. Hoang, R. Kumar, S. K. Panda, "Real-time Implementation of a Harmony Search Algorithm-based Clustering Protocol for Energy Efficient Wireless Sensor Networks", *IEEE Transactions on Industrial Informatics*, 2013.

Bibliography

[1] R. Verdone, D. Dardari, G. Mazzini and A. Conti, "Wireless Sensor and Actuator Networks: technologies, analysis and design", *Academic Press Elsevier* Chapter 1, 2008.

[2] Reinisch, C., Kastner, W., Neugschwandtner, G., Granzer, W., "Wireless Technologies in Home and Building Automation", *The 5th IEEE International Conference on Industrial Informatics*, 1, pp.93-98, 2007.

[3] Martinez, D., Blanes, F., Simo, J., Crespo, A., "Wireless Sensor and Actuator Networks: Charecterization and case study for confined spaces healthcare applications", *International Multiconference on Computer Science and Information Technology (IMCSIT 2008)*, pp.687-693, 2008.

[4] Tian H., Sudha K., Stankovic, J.A., Abdelzaher, T., Luo, L., Stoleru R., Ting Y., Lin G., Hui, J., and Krogh, B., "Energy-efficient surveillance system using wireless sensor networks", *In Proceedings of the 2nd international conference on Mobile systems, applications, and services (MobiSys '04)*, ACM, New York, USA, pp. 270-283, 2004

[5] J. Yick, B. Mukherjee, and D. Ghosal, Wireless Sensor Network Survey, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 52, no. 12, pp.: 2292-2330, Aug. 2008.

[6] I.F. Akyildiz, W.L. Su, S. Yogesh and C. Erdal, "A Survey on Sensor Networks", *IEEE Communications Magazine*, vol.40, no.8, pp.102114, 2002.

[7] Q. Li and C. Yao, "Real time concepts for Embedded Systems", *CMP Books*, Chapter 4, 2003

[8] http://www.cs.utsa.edu/~korkmaz/teaching/cs6543/

[9] Jae Myeong Choi, Heau-Jo Kang and Yong-Seok Choi, "A Study on the Wireless Body Area Network Applications and Channel Models", *Second International Conference on Future Generation Communication and Networking (FGCN '08)*, vol.2, pp.263,266, 13-15 Dec. 2008

[10] V. Handziski, J. Polastre, J. H. Hauer, C. Sharp, A. Wolisz, D. Culler, "Flexible Hardware Abstraction for Wireless Sensor Networks", *Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN '05)*, January 31-February 2, 2005

[11] M. Kuorilehto, M. Kohvakka, J. Suhonen, P. Hmlinen, M. Hnnikinen, T. D. Hamalainen, "Ultra-Low Energy Wireless Sensor Networks in Practice: Theory, Realization and Deployment", *John Wileys & Sons, Ltd* Chapter 10, 2007.

[12] H. Zimmermann, "OSI Reference Model", *IEEE Transactions on Communications*, vol. COMM-28(4), p425, April 1980.

[13] Mitsubishi Electric Research Laboratories, "Zigbee Security Toolbox", www.merl.com/areas/zigbeesec.

[14] Zigbee Alliance, www.zigbee.org/home.aspx.

[15] IEEE Standard 802.15.4, ”Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), September 2006.

[16] F. Dressler, ”Self-Organization in Sensor and Actor Networks”, *John Wiley & Sons*, Chapter 7, 2007.

[17] Jamal N. Al-karaki and Ahmed E. Kamal, ”Routing Techniques in Wireless Sensor Networks: A Survey”, *IEEE Wireless Communications*, vol. 11, pp. 6-28, 2004.

[18] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, ”Energy-efficient communication protocols for wireless microsensor networks”, *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS)*, Hawaii, USA, January 2000.

[19] Ramesh Rajagopalan and Pramod K. Varshney, ”Data aggregation techniques in sensor networks: A survey”, *IEEE Communications Surveys & Tutorials*, vol. 8, pp. 48-63, 2006.

[20] Yaacov Fernandess and Dahlia Malkhi, ”K-clustering in wireless ad hoc networks”, *Proceedings of the second ACM international workshop on Principles of mobile computing*, Toulouse, France, October 30-31, 2002

[21] L. Tan, Y. Gong and G. Chen, ”A balanced parallel clustering protocol for wireless sensor networks using K-means techniques”, *Proceedings of The Second International Conference on Sensor Technologies and Applications (SENSORCOMM 2008)*, Cap Esterel, France, pp. 25-31, August 2008.

[22] Edwin K. P. Chong, Stanislaw H. Zak, ”An introduction to optimization”, *Wiley-Interscience Publication*, 2nd Edition, 2004.

[23] Z.W. Geem, J.H. Kim and G.V. Loganathan, "A new heuristic optimization algorithm: harmony search", *Simulations* vol. 76 (2), pp. 60-68, 2001.

[24] ZongWoo Geem, "Music-Inspired Harmony Search Algorithm", *Springer*, 2008.

[25] K.S. Lee, Z.W. Geem, "A new meta-heuristic algorithm for continues engineering optimization: harmony search theory and practice", *Comput. Meth. Appl. Mech. Eng.*, 194, pp. 3902-3933, 2004.

[26] K.S. Lee, Z.W. Geem, *A new meta-heuristic algorithm for continues engineering optimization: harmony search theory and practice*, Comput. Meth. Appl. Mech. Eng. 194, pp. 3902-3933, 2004.

[27] Shah Hosseini, H., "Problem solving by intelligent water drops", *IEEE Congress on Evolutionary Computation (CEC)*, 2007

[28] Shah Hosseini, H., "The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm", *International Journal of Bio-Inspired Computation*, vol. 1, (1/2), pp. 71 - 79, Jan 2009.

[29] Haibin Duan and Senqi Liu and Xiujuan Lei., "Air robot path planning based on Intelligent Water Drops optimization", *IEEE IJCNN*, pp. 1397-1401, 2008.

[30] Shah Hosseini, H., "Optimization with the Nature-Inspired Intelligent water drops algorithm", *In Wellington Pinheiro dos Santos: 'Evolutionary Computation'*, Vienna, Intech, 2009.

[31] N.M. Abdul Latiff, C.C. Tsimenidis and B.S. Sharit, "Performance Comparison of Optimization Algorithms for Clustering in Wireless Sensor Networks",

IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS) Pisa, pp. 1-4, October 2007.

[32] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss and P. Levis., "Collection Tree Protocol", *ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2009.

[33] TinyOS, www.tinyos.net.

[34] Liao, W.H., Kao, Y., and Fan, C.M., "Data aggregation in wireless sensor networks using ant colony algorithm", *Journal of Network and Computer Applications* vol.31, No. 4, 2008.

[35] C. Chien, I. Elgorriaga, C.McConaghy, "Low-power directsequence spread-spectrum modem architecture for distributed wireless sensor networks", *International Symposium on Low Power Electronics and Design (ISLPED)*, Huntington Beach, California, August 2001.

[36] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks", *Proceedings of ACM MobiCom01*, pp. 272286, Rome, Italy, July 2001.

[37] C.C. Enz, A. El-Hoiydi, J.-D. Decotignia, V. Peiris, "WiseNET: an ultralow-power wireless sensor network solution", *IEEE Computer Society Press* vol. 37, pp. 6270, Los Alamitos, CA, USA, 2004.

[38] A.Y. Wang, S. Cho, C.G. Sodini, A.P. Chandrakasan, "Energy efficient modulation and MAC for asymmetric RF microsensor systems", *International Symposium on Low Power Electronics and Design (ISLPED)*, Huntington Beach, California, August 2001.

[39] S. Cui, A.J. Goldsmith, A. Bahai, "Energy-constrained modulation optimization", *IEEE Transactions on Wireless Communication* vol. 4, pp. 23492360, 2005.

[40] J. Ammer, J. Rabaey, "The energy-per-useful-bit metric for evaluating and optimizing sensor network physical layers", *IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, 2006.

[41] H. Karl and A. Willig, "Protocols and Architectures for Wireless Sensor Networks", *John Willy & Sons, Ltd*, 2005.

[42] Wei Ye and John Heidemann and Deborah Estrin, "An Energy-Efficient MAC protocol for Wireless Sensor Networks", *Proceedings of the IEEE Infocom*, New York, NY, USA, June 2002.

[43] van Dam T. and Koen Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks", *Proceedings of the International Conference ON Embedded Networked Sensor Systems (Sensys'03)*, Los Angeles, CA, USA, 2003.

[44] Lu, G.; Krishnamachari, B.; Raghavendra, C.S.; , "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," *Proceedings of the 18th InternationalParallel and Distributed Processing Symposium*, pp. 224, 26-30 April 2004.

[45] Rajendran, Venkatesh and Obraczka, Katia and Garcia-Luna-Aceves, J. J., "Energy-efficient collision-free medium access control for wireless sensor networks", *Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys)*, Los Angeles, CA, USA, 2003.

[46] Sohrabi, K.; Gao, J.; Ailawadhi, V.; Pottie, G.J.; , "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol.7, no.5, pp.16-27, Oct 2000.

[47] Wendi B. Heinzelman and Anantha P. Ch and IEEE and Anantha P. Chandrakasan and Member and Hari Balakrishnan and and Hari Balakrishnan., "An Application-Specific Protocol Architecture for Wireless Microsensor Networks", *IEEE Transactions on Wireless Communications* vol. 1, pp. 660-670, 2002.

[48] Singh, Suresh and Raghavendra, C. S., "PAMASpower aware multi-access protocol with signalling for ad hoc networks", *ACM SIGCOMM Computer Communication Review*, vol. 28, No. 3, pp. 5-26, New York, USA, 1998.

[49] Schurgers, C., Tsiatsis, V. and Srivastava, M.B. , "STEM: Topology management for energy efficient sensor networks," *IEEE Aerospace Conference Proceedings*, vol.3, no., pp. 1099-1108, 2002.

[50] Joseph Polastre, Jason Hill and David Culler, "Versatile low power media access for wireless sensor networks", *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys)*, New York, NY, USA, 2004

[51] Kai-Juan Wong and D. K. Arvind, "SpeckMAC: low-power decentralised MAC protocols for low data rate transmissions in specknets" *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality REALMAN* New York, NY, USA, 2006

[52] Injong Rhee, Warrier, A., Aia, M., Jeongki Min and Sichitiu, M.L., "Z-MAC:

A Hybrid MAC for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol.16, no.3, pp.511-524, June 2008.

[53] MEMSIC, "TelosB mote platform", *http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=152%3Atelosb*

[54] MEMSIC, "MICA Mote platform", *http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=147%3Amica2*

[55] MEMSIC, "IRIS wireless sensor motes", *http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=135%3Airis*

[56] J. Polastre, R. Szewczyk, and D. Culler., "Telos: Enabling ultra-low power wireless research", *In Proc. IPSN/SPOTS05*, Los Angeles, CA, USA, April 2005.

[57] EYES, the EU Project EYES (IST 2001 34734), *http://Eyes.eu.org*, 2005.

[58] K. J. Wong, D. K. Arvind, "Speckled Computing: Disruptive Technology for Networked Information Appliances", *In Proceedings of the IEEE International Symposium on Consumer Electronics (ISCE'04)*, pp. 219-223, UK, September 2004.

[59] S. Hedetniemi, A. Liestman, "A survey of gossiping and broadcasting in communication networks", *IEEE Networks* vol. 18, No. 4, p319349, 1988.

[60] W. Heinzelman, J. Kulik, H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks", *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, Seattle, WA, August 1999.

[61] C. Intanagonwiwat, R. Govindan, D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks", *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00)*, Boston, MA, August 2000.

[62] R. Shah, J. Rabaey, "Energy aware routing for low energy ad hoc sensor networks", *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Orlando, FL, March 2002.

[63] T. Kang , J. Yun , H. Lee, I. Lee, H. Kim, B. Lee, K. Han, "A Clustering Method for Energy Efficient Routing in Wireless Sensor Networks", *Proceeding of the 6th WSEAS International Conference on Electronics, Hardware, Wireless and Optical Communications*, 2007.

[64] R. Wang, G. Liu, C. Zheng, "A Clustering Algorithm Based on Virtual Area Partition for Heterogeneous Wireless Sensor Networks", *Proceeding of International Conference on Mechatronics and Automation*, 2007.

[65] K. Shin, A. Abraham, S. Han, "Self organizing sensor networks using intelligent clustering", *Lecture Notes in Computer Science*, p40-49, 2006.

[66] H. Chan, A. Perrig, "ACE: An Emergent Algorithm for Highly Uniform Cluster Formation", *European Workshop on Sensor Networks*, p154-171, 2004.

[67] S. Lindsey and C. Raghavendra, "PEGASIS: Power-Efficient Gathering in

Sensor Information Systems", *IEEE Aerospace Conference Procedings*, vol. 3, 916, pp. 112530, 2002.

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

[69] A. Manjeshwar and D. P. Agarwal, "APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks", *Proceedings of the 2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile computing*, Ft. Lauderdale, FL, April 2002.

[70] J.S. Liu, C.H.R Lin, "Energy-efficiency clustering protocol in wireless sensor networks", *Ad Hoc Networks*, 2005.

[71] Y. Xu, J. Heidemann, D. Estrin, "Geography-informed energy conservation for ad hoc routing", *Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'01)*, Rome, Italy, July 2001.

[72] Y. Yu, D. Estrin, R. Govindan, "Geographical and energyaware routing: a recursive data dissemination protocol for wireless sensor networks", *UCLA Computer Science, Department Technical Report, UCLA-CSD TR-01-0023*, May 2001.

[73] https://sites.google.com/site/dataclusteringalgorithms/home

[74] Fung, G., "A Comprehensive Overview of Basic Clustering Algorithms" http://pages.cs.wisc.edu/~gfung/clustering.pdf>, 2001.

[75] Hartigan J. A. and Wong M. A., " A K-Means Clustering Algorithm", *Applied Statistics*, vol 28, No. 1, pp. 100-108, 1979.

[76] MacQueen J., "Some methods for Classification and Analysis of Multivariable Observations", *Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281-297, University of California Press, Berkeley, 1967.

[77] Bezdek, J.C, "Pattern Recognition with Fuzzy Objective Function Algorithms", *Plenum Press*, New York, 1981.

[78] Zhuang, S., Yan Huang, Palaniappan, K., Yunxin Zhao, "Gaussian mixture density modeling, decomposition, and applications," *IEEE Transactions on Image Processing*, vol. 5, no. 9, pp. 1293-1302, Sep 1996

[79] Rui Xu and Wunsch, D., II, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645-678, May 2005.

[80] Kulkarni, R.V., Forster, A., Venayagamoorthy, G.K., "Computational Intelligence in Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol.13, no.1, pp.68-96, First Quarter 2011

[81] Kulkarni, R.V. and Venayagamoorthy, G.K., "Particle Swarm Optimization in Wireless-Sensor Networks: A Brief Survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol.41, no.2, pp.262,267, March 2011

[82] Carter, Michael., "Foundations of mathematical economics," *The MIT Press*, 2001.

[83] K. De Jong, "Evolutionary computation: a unified approach", *MIT Press*, Cambridge, 2006.

[84] J.H. Holland, "Adaptation in natural and artificial systems", *The University of Michigan Press*, Ann Arbor, 1975.

[85] D.B. Fogel, "What is evolutionary computation?", *IEEE Spectrum*, vol. 37, pp. 26-32, 2000.

[86] David E. Goldberg, "Genetic Algorithms in search, Optimization, and Machine Learning", *Preason Education*, 9th Edition, 2005.

[87] M. Mitchell, "An introduction to genetic algorithms", *Prentice Hall*, 1st Edition, India, 1996

[88] J. Kennedy and R.C. Eberhart, "Particle swarm optimization", *Proceedings of IEEE International Conference on Neural Networks*, pp. 1942-1948, 1995.

[89] Y. Dong, B.X.J. Tang and D. Wang, "An application of swarm optimization to nonlinear programming", *Computers & Mathematics with Applications* vol. 49 (11-12), pp. 1655-1668, 2005.

[90] I. Montalvo, J. Izquierdo, R. Prez, and P.L. Iglesias, "A diversity-enriched variant of discrete PSO applied to the design of Water Distribution Networks", *Engineering Optimization*, 2007.

[91] Andreas Konstantinidis, Kun Yang, Qingfu Zhang, and Demetrios Zeinalipour-Yazti, "A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks", *Computer Networks* vol. 54, 6, pp. 960-976, April 2010.

[92] Jie Jia, Jian Chen, Guiran Chang, and Zhenhua Tan, "Energy efficient coverage control in wireless sensor networks based on multi-objective genetic al-

gorithm" *Computers and Mathematics with Applications*, vol. 57, 11-12, pp. 1756-1766, June 2009.

[93] Anna L. Buczak, Henry (Hui) Wang, Houshang Darabi, and Mohsen A. Jafari, "Genetic algorithm convergence study for sensor network optimization", *Information Sciences*, vol. 133, 3-4, pp. 267-282, April 2001.

[94] T. Wimalajeewa and S. K. Jayaweera, "Optimal Power Scheduling for Correlated Data Fusion in Wireless Sensor Networks via Constrained PSO", *IEEE Transaction on Wireless Communications*, vol. 7, 9, pp. 3608-3618, September 2008.

[95] Yang, Xin-She. "Harmony search as a metaheuristic algorithm.", *In Music-inspired harmony search algorithm, Springer Berlin Heidelberg*, pp. 1-14, 2009.

[96] Al-Karaki, J., Ul-Mustafa, R., and Kamal, A., "Data aggregation in wireless sensor networks - exact and approximate algorithms", *Workshop on High Performance Switching and Routing (HPSR)*, 2004.

[97] Krishnamachari, L., Estrin, D., and Wicker, S., "The impact of data aggregation in wireless sensor networks", *Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops*, 2002.

[98] Intanagonwiwat, Estrin, D., C., Govindan, R., and Heidemann, J., "Impact of network density on Data Aggregation in wireless sensor networks" *Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops*, 2002

[99] Misra, R., and Mandal, C., "Ant-aggregation: ant colony algorithm for optimal data aggregation in wireless sensor networks", *International Conference on Wireless and Optical Communications Networks (IFIP)*, 2006.

[100] Marco Dorigo and Luca Maria Gambardella., "Ant Colony System: A cooperative learning approach to the traveling salesman problem", *IEEE Transations on Evolutionary Computation* vol. 1, No. 1, pp. 53-66, 1997.

[101] Marco Dorigo and Mauro Birattari and Thomas Sttzle., "Ant Colony Optimization – Artificial Ants as a Computational Intelligence Technique", *IEEE Computer Intelligence Magazine*, vol. 1, pp. 28-39, 2006.

[102] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. "Wireless Sensor Networks for Habitat Monitoring", *In Proceedings of the 1st ACM Workshop on Wireless Sensor Networks and Applications*, Atlanta, GA, September 2002.

[103] A. Sinha, Energy-Scalable Software, *S.M. Thesis, Department of EECS, Massachusetts Institute of Technology*, February 2000.

[104] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks", *Proceedings of ACM MobiCom01*, Rome, Italy, pp. 272286, July 2001.

[105] M. Kangas, A. Konttila, I. Winblad and T. Jms, "Determination of simple thresholds for accelerometry-based parameters for fall detection", *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2007*, pp.1367-1370, 2007.

[106] D.M. Karantonis, M.R. Narayanan, M. Mathie, N.H. Lovell and B.G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring", *IEEE Transactions on Information Technology in Biomedicine*, vol.10, issue.1, pp.156-167, 2006.

[107] I. Stark, "Invited Talk: Thermal Energy Harvesting with ThermoLife", *International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2006)*, pp.19-22, 2006.

[108] Y.K. Tan and S.K. Panda, "A Novel Piezoelectric Based Wind Energy Harvester for Low-power Autonomous Wind Speed Sensor", *33th Annual IEEE Conference of Industrial Electronics Society (IECON07)*, pp.2175-2180, 2007.

[109] M. Broussely and G. Pistoia, "Industrial Applications of Batteries From Cars to Aerospace and Energy Storage", *Elsevier*, 2007.

[110] The Samraksh Company, "Users Manual for the BumbleBee (Model 0) A Mote-Scale Pulsed Doppler Radar for use in Wireless Sensor Networks", http://www.samraksh.com/docs/BumbleBee-UM-v103.pdf.

[111] Kemal Akkaya and Mohamed Younis. "A survey on routing protocols for wireless sensor networks", *Ad Hoc Networks*, vol. 3, pp. 325-349, 2005.

[112] M. Ettus, "System capacity, latency, and power consumption in multihop-routed SS-CDMA wireless networks", *Radio and Wireless Conference (RAWCON)*, pp. 55-58, August 1998.

[113] Rengasamy, M., Dutkiewicz, E., Hedley, M., "MAC Design and Analysis for Wireless Sensor Networks with Co-operative Localisation", *International Symposium on Communications and Information Technologies ISCIT* pp. 942-947, Oct 2007.

[114] Kim, Sang-Chul, Woo, Chong-Woo, "A Scalability Analysis of TDMA-Based Ad Hoc MAC Protocols", in Gervasi, O., Taniar, D., Murgante, B., Lagan, A., Mun, Y., Gavrilova, M., "Computational Science and Its Applications

ICCSA", *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 580-592, 5592, 2009.

[115] *MTS/MDA Sensor Board User's Manual*, Crossbow. Revision A, June 2007.

[116] W. B. Heinzelman, "Application-Specifc Protocol Architectures for Wireless Networks", Ph.D. dissertation, Deptt of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, US, 2000.