ENERGY-EFFICIENT ROUTING ALGORITHMS FOR WIRELESS SENSOR NETWORKS

BARRA TOURAY

A thesis submitted in partial fulfilment of the requirements of the Liverpool John Moores University for the degree of Doctor of Philosophy

School of Engineering, Technology and Maritime Operations,
Liverpool John Moores University

JUNE 2013

"Read in the name of your Lord who created man out of clots of congealed blood. Read, for your Lord is the most generous. He who taught the use of the pen that man might be taught that which he did not know."

Holy Quran

Narrated Abu Huraira:

The Prophet said, "Religion is very easy and whoever overburdens himself in his religion will not be able to continue in that way. So you should not be extremists, but try to be near to perfection and receive the good tidings that you will be rewarded; and gain strength by worshipping in the mornings, the nights."

Prophet Teachings

Dedicated to:
My beloved parents, kids
and wife

ENERGY EFFICIENT ROUTING ALGORITHMS FOR WIRELESS SENSOR NETWORKS

by

Barra Touray

Abstract

A wireless sensor network (WSN) is made of tiny sensor nodes usually deployed in high density within a targeted area to monitor a phenomenon of interest such as temperature, vibration or humidity. The WSNs can be employed in various applications (e.g., Structural monitoring, agriculture, environment monitoring, machine health monitoring, military, and health). For each application area there are different technical issues and remedies. Various challenges need to be considered while setting up a WSN, including limited computing, memory and energy resources, wireless channel errors and network scalability.

One way of addressing these problems is by implementing a routing protocol that efficiently uses these limited resources and hence reduces errors, improves scalability and increases the network lifetime. The topology of any network is important and wireless sensor networks (WSNs) are no exception. In order to effectively model an energy-efficient routing algorithm, the topology of the WSN must be factored in. However, little work has been done on routing for WSNs with regular patterned topologies, except for the shortest path first (SPF) routing algorithms. The issue with the SPF algorithm is that it requires global location information of the nodes from the sensor network, which proves to be a drain on the network resources. In this thesis a novel algorithm namely, BRALB (Biased Random Algorithm for Load Balancing) is proposed to overcome the issues faced in routing data within WSNs with regular topologies such as square-base topology and triangle-based topology. It is based on

random walk and probability. The proposed algorithm uses probability theory to build a repository of information containing the estimate of energy resources in each node, in order to route packets based on the energy resources in each node and thus does not require any global information from the network. It is shown in this thesis by statistical analysis and simulations that BRALB uses the same energy as the shortest path first routing as long as the data packets are comparable in size to the inquiry packets used between neighbours. It is also shown to balance the load (i.e. the packets to be sent) efficiently among the nodes in the network. In most of the WSN applications the messages sent to the base station are very small in size. Therefore BRALB is viable and can be used in sensor networks employed in such applications. However, *one of the constraints of BRALB* is that it is *not very scalable*; this is a genuine concern as most WSNs deployment is large scale.

In order to remedy this problem, C-BRALB (Clustered Biased Random Algorithm for Load Balancing) has been proposed as an extension of BRALB with clustering mechanism. The same clustering technique used in Improved Directed Diffusion (IDD) has been adopted for C-BRALB. The routing mechanism in C-BRALB is based on energy biased random walk. This algorithm also does not require any global information apart from the initial flooding initiated by the sink to create the clusters. It uses probability theory to acquire all the information it needs to route packets based on energy resources in each cluster head node. It is shown in this thesis by using both simulations and statistical analysis that C-BRALB is an efficient routing algorithm in applications where the message to be sent is comparable to the inquiry message among the neighbours. It is also shown to balance the load (i.e. the packets to be sent) among the neighbouring cluster head nodes.

Acknowledgement

It has been a great privilege and honour to be associated with ***Dr. Princy Johnson, Dr. Martin Randles and Prof. Emil Levi*** for carrying out the research work towards my PhD degree. Their professional guidance, invaluable advice, patience, constant support and encouragement have made this thesis possible. Therefore, I would like to take this opportunity to express my sincere gratitude and thanks to all of them.

Some parts of this work would not have been possible without the active contribution of ***Jinn Shim, Muhammad Tabassum Tahir, Veera Venkata Satyaphanidar Kotti*** and ***Dr Eammanual Jammeh*** and for that, I am very grateful.

I am also grateful to ***The British Council*** for funding my PhD studies through their ***Common Wealth Scholarship Scheme.*** My sincere gratitude goes to the entire staff of the ***Engineering School of Liverpool John Moores University*** especially the school Director ***Professor Ian Jenkinson, Dr Mike Shaw*** deputy school director, ***Valerie Mizen*** secretary and ***Dr Princy Johnson*** for sponsoring this work before I got the scholarship from the British Council. My sincere thanks and gratitude also go to Gamtel, all my teachers, friends and uncles for their assistance in making this thesis a success.

A special thanks to my ***father Chief Baboucarr Touray***, my ***mother Aji Ramoutoulie Njie***, *my lovely* **wife Haddijatou Jallow**, my ***daughters Aminata Touray and Ramatoulie Touray*** and my ***son Baboucarr Touray*** for their endless support and motivation throughout my Ph.D studies. Without their

constant assurance and assistance, completion of this research work would have not been possible. And as a sign of my love, gratitude and affection I dedicate this thesis to them appreciating their immense contributions towards the success of my PhD thesis.

Thank you all,

Barra Touray.

List of Tables

List of Figures

List of Abbreviations and Symbols

AACOCM	Advanced Ant Colony algorithm based On Cloud Model
ACOA	Ant Colony Optimization Algorithm
ADC	Analogue Digital Converter
ADV	New data ADVertisement
AODV	Ad hoc On-Demand Distance Vector
API	Application Programming Interface
ARA	Ant Routing Algorithm
BAN	Body Area Network
BN	Border Node
BRALB	Biased Random Algorithm for Load Balancing
BS	Base Station
C-BRALB	Clustered Biased Random Algorithm for Load Balancing
CBRP	Clustered Based Routing Protocol
CCHN	Candidate Cluster Head Node
CCP	Coverage Configuration Protocol
CH	Cluster Head
CHN	Cluster Head Node
CMN	Cluster Member Node
CMOS	Complementary Metal–Oxide–Semiconductor
CPU	Central Processing Unit
CSMA	Carrier Sense Multiple Access
DD	Directed Diffusion
DDBCI	Directed Diffusion Based on Clustering and Inquiry
DSP	Digital Signal Processing

DTN	Delay Tolerant Network
EAR	Eavesdrop-And-Register algorithm
EIGRP	Enhanced Interior Gateway Routing Protocol
EPACOR	Energy Prediction and Ant Colony Optimization Routing
FDM	Frequency division Multiplexing
GDI	Great Duke Island
GPS	Global Positioning System
HEED	Hybrid Energy Efficient Distributed
IC	Integrated Circuit
ID	Identifier
IDD	Improved Directed Division
IEEE	Institute of Electrical and Electronics Engineers
IGRP	Interior Gateway Routing Protocol
IP	Internet Protocol
IS-IS	Intermediate System to Intermediate System
KJ	Kilojoules
K-NC	K-Non-unit-disk Coverage
K-UC	K-unit-disk coverage
LEACH	Low-Energy Adaptive Clustering Hierarchy
LREP	Location REPLY packet
LREQ	Location Request Packet
MAC	Media Access Control
MANET	Mobile Ad-hoc Networks
MCFA	Minimum Cost Forwarding Algorithm
MTPR	Minimum total Transmission Power-Routing
NS-2	Network Simulator 2

ORN	Ordinary Node
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PAN	Personal Area Network
PCA	Probabilistic Coverage Algorithm
PDA	Personal Digital Assistant
PEGASSI	Power-Efficient Gathering in Sensor Information Systems
PERA	A Probabilistic Emergent Routing Algorithm for Mobile Ad Hoc Networks
PL	Path Loss
PLL	Phase Locked Loop
QoS	Quality of Services
REB-R	Remaining-Energy Based Routing
RF	Radio Frequency
RIP	Routing Information Protocol
RSS	Road Side Sensors
SMACS	Medium Access Control for Sensor Networks
SN	Sensor Node
SPF	Shortest Path First
SPIN	Sensor Protocols for Information via Negotiation
SPIN-1	Sensor Protocols for Information via Negotiation-1
T-ANT	TCCA Ant
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TTL	Time To Live
UCB	University of California at Berkeley

UDP	User Datagram Protocol
VASENET	Vehicular Ad hoc and SEnsor NETworks
VCO	Voltage-Controlled Oscillator
WBAN	Wireless Body Area Network
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Networks

Table of Contents

Chapter 1

INTRODUCTION

Chapter 1

INTRODUCTION

1.1 Background

With recent developments in wireless technology every location on the earth's surface has become accessible. People all over the world rely on wireless communication to exchange data using pagers, telephones, laptops and various personal digital assistants (PDAs) and other wireless communication products. The rapidly expanding uptake of wireless voice and messaging services has paved the way for wireless communication to be applied to many areas of application, including personal and business computing, and military surveillance. With many of the limitations of static wired networks overcome, people are now able to access and share information all over the world especially in situations where it would have been previously impossible.

It is developments such as these that have made it possible to easily create Ad Hoc Networks in situations where a wired network would have been difficult or even impossible, to implement. These advancements, together with the recent innovations in miniaturisation, simple cost-effective, low power circuit design, and small-size batteries, have made a new technological vision possible: Wireless Sensor Networks (WSN) (Estrin et al. 1999; Kahn et al. 1999; Akyildiz et al. 2002; Jain and Agrawal 2005; Cordeiro and Agrawal 2006).

WSN are a special form of Ad Hoc Networks. They are different from Mobile Ad-hoc Networks (MANETs) in as much as they are limited in mobility and even in instances of mobile functionality are much slower than the MANETs. WSN networks combine wireless communication and limited computation facilities that can sense physical phenomena, such as temperature, pressure or vibration, and can easily be embedded in our physical environment (Cordeiro and Agrawal 2006).

WSN has many uses especially in military applications, such as battle field surveillance, which is one of the key motivations for the observed rapid development in this field. WSNs are also used in industrial and civilian application areas, such as environment and habitat monitoring, healthcare applications, process monitoring and control, home automation, and traffic control.

WSNs can be densely deployed in the order of thousands of nodes over a small area. The sensor nodes are low powered and are equipped with radio transmission or other wireless communications devices. They have multiple parameter sensing units, minute microcontrollers, and an energy source usually in the form of a non-rechargeable battery. These developments made it possible to take advantage of the benefits that WSN have over equivalent wired networks. Some of these advantages are: ease of deployment, fault tolerance, extended range and mobility. However, wireless media have a few inherent limitations such as limited bandwidth, error prone transmissions, susceptibility to electrical interference and potential collisions during channel access. Once the nodes have been randomly deployed it becomes very difficult or even impossible to replace the individual nodes or batteries. Therefore, the protocols designed for such networks must strategically distribute the dissipation of energy in order to enhance the lifetime of the network (Cordeiro and Agrawal 2006). A sensor network should also have some additional features such as attribute-based addressing, location awareness and query handling.

There are various applications for WSN; however, these come with various challenges and restrictions. WSNs have limited computing power, limited energy supply and limited bandwidth of the wireless link connecting the nodes. Therefore the design of a successful WSN must maximise the energy level of the individual sensor nodes, without compromising much of the limited computing power and the limited bandwidth. One way of dealing with this is by defining and deploying energy-efficient routing protocols for WSN. It is for this reason that much research is conducted in this area. Routing protocols in WSN are influenced by many factors which must be dealt with, before efficient communication can be achieved. Some of these challenging factors are:

- Energy consumption without losing accuracy
- Computational capabilities
- Ad Hoc deployment
- Scalability
- Communication range
- Connectivity and transmission media
- Fault tolerance and QoS
- Security and control overhead

In this thesis, a novel Biased Random Algorithm for Load Balancing (BRALB) in Wireless Sensor Networks for *environment monitoring* is proposed. BRALB is further modified and integrated with a clustering scheme to increase its efficiency and resources utilisation. The resulting algorithm is called C-BRALB. The proposed algorithms are statistically analysed and then simulated using *NETLOGO* and *NS-2* modelling software packages. The generated network system provides an effective, energy efficient, scalable, and reliable non-centralised routing algorithm for Wireless Sensor networks.

1.2 Research Motivation

One of the critical design issues for WSN is how to conserve energy. The main components that consume energy in WSN are: *Communication Unit* (transmission and receiver radio), *Sensing* Unit (Sensing transducer and A/D (Analogue-digital converter)), and Computing/ Processing Unit. These key elements that consumed the major energy in a wireless sensor network are built-into the sensor node; therefore the only way to improve the energy problem is to use it efficiently. One method of conserving the energy of a WSN is by using an energy efficient routing protocol.

The traditional routing protocols for wired networks such as EIGRP, OSPF, ISIS, IGP, IGRP and RIP are not well suited for WSN. The MANET routing protocols are also not suited for WSN, due to the energy constraint in WSNs, large node deployment and computational power among other things. This brought the need for novel routing protocols that are specifically designed for WSN with all the key characteristics and features of WSN taken into consideration.

This urgent need for bespoke routing protocols for WSN has paved way for new routing protocol designs. The design of any successful routing protocol must take into account the topology of the network. In WSN different applications require different topologies and hence different routing and scheduling algorithms. Traditionally, most networks were analysed as random networks, but recent findings have proved that, in practice, networks exhibit features that go beyond randomness. Most networks exhibit features that are far from being random, displaying characteristics such as *preferential attachment*, *large clustering coefficients* and *self-organising capabilities*. Therefore there is a need to revisit our networks and analyse them thoroughly to *develop appropriate statistical models* for the networks.

One issue highlighted in clustering algorithm is the determination of the best value of a cluster size for a given wireless sensor network. It is assumed that when a cluster size is *bound to k-hop communication*, *smaller number of clusters is better*. This could be true to some extent depending on the actual cluster size. If the *cluster sizes are too big* it can result in *additional overhead due* to *too many intra-cluster communications*, resulting in inefficiency. However, there is no investigation of how to identify the right cluster size (Förster et al. 2010). This has been investigated statistically and the formula for obtaining the right cluster size has been derived for C-BRALB in this thesis. This equation can be used to have a rough estimation of the best cluster size for a given network size.

Generally, there are two evaluation scenarios to successfully obtain the true behaviour and performance of a new algorithm. These are called fixed network evaluation and scalability analysis. In fixed network evaluation method, some parameters such as the network, the application, the data traffic, etc. are fixed and used to evaluate and compare the targeted protocols in terms of network lifetime, energy expenditure, incurred routing overhead etc. This evaluation method is only valid for comparative analysis. On the other hand, measuring some of these properties might be meaningless, i.e., it is not possible to evaluate that the reported network lifetime for protocol X is sufficient or not (Förster et al. 2010).

The solution to the above mentioned issue is the scalability analysis which independently analyses a protocol showing its properties and behaviour with various network settings, like number of nodes, network size, data traffic etc. Under such circumstances comparative analysis might be misleading (Förster et al. 2010). For example, the energy expenditure of some routing protocol may skyrocket with increasing data while the new protocol may show slightly lower energy expenditure. The interpretation of this result could be that the new protocol is scalable while on the other hand it could mean that both protocols are not scalable hence the confusion. Hence there is a need for *isolated evaluation* of the new protocol.

Unfortunately the reality is that most protocols are evaluated using only one approach. This short coming is addressed in this thesis by using both methods to evaluate BRALB and C-BRALB. The statistical models ought to reveal the true features of these networks and can be used as an input in the design of a novel routing or scheduling algorithm.

WSN have other characteristics and these also need to be factored into the design. These features are:

- Due to the large density of nodes in WSN, which can be on the order of thousands of nodes per km^2, *sensor nodes do not have unique addresses*, as this will mean increased overhead.
- Most applications of WSN require the nodes to be stationary. However, in some instances the nodes may move and change location with limited mobility.
- Sensor nodes should be self-organising as the operation of a sensor network is usually unattended; the network must organise and reconfigure itself when failure occurs as a result of node failure or other issues.
- The positions of sensor nodes are important as data collection is based on it. Triangulation method may be used as an approximate sensor location using radio strength from a few known points (Bulusu et al. 2000). Algorithms using triangulation work well in conditions where only few nodes know their positions a priori, e.g. using GPS hardware (Bulusu et al. 2000; Cordeiro and Agrawal 2006).
- The design of WSN networks is application specific therefore the challenging problem for a periodic weather-monitoring task, for instance, is different from a time constrained military application such as precision tactical surveillance.
- Sensor networks are data centric, that is, data is requested based on attributes such as attribute based addressing while in traditional networks data is requested from a particular node. In attribute based addressing a query can be something like humidity above a certain value. The nodes that measure humidity, above that value, need to respond to the query.
- Data collected from sensor is based on common phenomenon with a high probability that nodes close to each other will be collecting the same data. With this, there is the need for *data aggregation techniques* so as to have *energy efficient data delivery*.

1.3 Research Aim and Objectives

WSN is a network of partially distributed autonomous devices that depend on sensors to monitor environmental or physical conditions. The sensors are usually powered by non-rechargeable batteries and are usually distributed randomly in large numbers all over the network and as such there is the problem of how to efficiently route collected data through the WSN.

The main problem, in a wireless sensor network is how to conserve energy. Node clustering is one way of trying to maximise the energy of nodes in WSN. In a WSN most of the time the nodes are densely deployed and therefore to save energy there is a need to cluster the nodes and coordinate the data transmission so as to avoid transmitting redundant data. Due to the random deployment of sensor nodes and the physical environment, there is a tendency for similar data to be generated in the same area. All this encourages clustering so that data can be aggregated within a cluster and then transmitted. This reduces both data transmission and contention within the WSN.

There are two major types of cluster formation namely grid clustering and random clustering. To summarise, *the aim of this research* is to p*ropose and develop a set of routing and clustering algorithms* for *randomly deployed WSNs*.

The routing policy in the WSN to a great extent affects the network life time. The number of hops a packet has traversed to reach its destination shows the energy consumed in transmitting that packet. The more nodes a packet passes before reaching its destination, the more energy is consumed in that network. If a particular route is used consistently more than the others, then the energy of the nodes in that route will be used up faster and hence they die sooner than the others. This is undesirable as it may bring partitions within the WSN and hence some areas will be isolated and therefore not monitored. In order to avoid such a problem, power-aware route selection mechanisms are needed that will be able to route messages along all possible routes from source to destination.

The main objectives of this research work are:

- To review routing algorithms and clustering schemes for WSNs.
- To develop a set of novel biased random routing algorithms for a specific WSN (environmental monitoring) application.
- To model and design a novel clustering scheme for WSN that links with the developed random routing protocol.
- To implement the routing algorithm and the clustering scheme on a complex network environment and analyse the results.

1.4 Contributions

The field of routing algorithm for Wireless Sensor Networks has been active for decades and many techniques and problem formulations have been used. However, implementing an efficient routing mechanism is a challenging research area. There is a need to consider the limited resource of a sensor network, the network size, the mode of deployment (Random or Regular) and the application. The application is very important as different applications require different level of network parameters such as delay sensitivity, security, data size,.

To factor these essential parameters, a novel energy efficient routing mechanism that provides a distributed load balancing mechanism is proposed in this thesis. This novel routing mechanism leads to decentralised, self-organised, and scalable *network* that depends only on local information for routing and routing update. Since the resulting network system (in C-BRALB) is randomly generated, it will be more resilient to *random errors*.

A novel energy efficient routing protocol based on modified random walk has been developed and then further incorporated with clustering that is suitable for use in both small-scale and large-scale networks (Touray et al. 2011). A clustering scheme for load distribution which uses local knowledge to distribute and balance the load in the network has been derived. Analytical and simulation results have been used to evaluate and validate the proposed clustering scheme.

Though *similar techniques exist* for routing mechanism, *the proposed scheme has the advantage* of using *a simple routing scheme based on random walk* and *local knowledge to efficiently distribute the load*. Moreover, this routing mechanism *does not require any global knowledge* apart from the initial flooding to create the clusters since its routing decision is based on random walk and local knowledge.

The proposed routing scheme is an energy efficient, scalable and load balancing algorithm. While other random walk load-balancing algorithms have been proposed in the literature (Tian et al. 2005; Rachuri 2009), most of them are not very scalable (Tian et al. 2005; Shokrzadeh 2007). This has been addressed in this thesis by the use of clustering which renders its scalability to C-BRALB.

In addition, the random forwarding technique is improved by biasing the forwarding of data toward neighbour nodes with highest remaining energy instead of choosing them uniformly at random. Hence, the nodes' selection criteria will be based on a predefined criterion rather than selecting any neighbour at random.

There are various parameters used to evaluate clustering schemes. One common approach is the use of energy expenditure or communication overhead to measure efficiency. However, most protocols have been evaluated based on the number of clusters or cluster heads, with the interpretation of low number of clusters as a good result. This can lead to the problem of too much in-cluster communications resulting in loads of in-cluster routing overhead. This shortcoming has also been addressed in this thesis.

Therefore, a novel energy biased routing algorithm is proposed that 'optimises' the limited memory and energy resources in WSNs and thus enabling the network to have a longer lifetime.

1.5 Thesis Structure

The thesis is divided into eight chapters. Chapter 1 is an introductory chapter that outlines the research problem undertaken, WSNs and its applications. Details of the scope of this research are also given here. Chapter 2 presents the relevant literature review related to Wireless Sensor networks concepts and discusses various concepts and standards in this area. In Chapter 3, the concept and modelling of Wireless Sensor network and its application are given. Chapter 4 discusses the routing techniques used to route data in wireless sensor networks. Chapter 5 proposes a novel biased random algorithm for load balancing (BRALB) in wireless sensor networks. Moreover, the analytical solution for the efficiency of this algorithm is presented here. Performance evaluation of the proposed algorithm and the *research results* are reported in Chapter 6. In addition, the new clustering scheme incorporated with BRALB creating Clustered Biased Algorithm for Load Balancing (C-BRALB) is presented here. In Chapter 7, conclusions are drawn from the research described in this thesis. Also, suggestions for future work are outlined in this chapter.

Chapter 2

LITERATURE REVIEW

Chapter 2

LITERATURE REVIEW

2.1 Introduction

Recent developments in science and technology have paved way to the emergence of WSNs. A WSN is an event-based system that uses several sensor nodes. The sensors are usually deployed in a random manner which may lead to coverage and connectivity issues. Therefore it is important to have full connectivity and coverage for a fully functioning WSN. However, connectivity and coverage is not only affected by node deployment method but also by the switching off of sensor nodes in order to avoid battery drain. One way of maximising the network lifetime is the use of energy efficient routing protocols which operate at the network layer. The network layer is responsible for the operation of the network, packet routing and flow control. However, it is often assumed that flow control is of little significance to the WSNs due to their limited data throughput and sometimes lack of quality of services (QoS)requirements (Callaway 2003). However, energy efficient protocols do not come free and there is a trade-off between energy efficiency and delay performance.

2.2 Review of coverage issues in WSNs

Sensing is a vital part of WSNs. For efficiency, it is important to use the minimum number of sensors possible to cover the sensing area in order to reduce cost. There are three general models for defining coverage issues in WSN, with random node deployment. The binary model models each sensor coverage area like a disk. Any activity taking place within the disk is monitored by a sensor located at the centre of the disk; otherwise it is not monitored by the sensor. The probabilistic model is the second one. In this model an event happening within the coverage of a sensor network may or may not be detected depending on the probability distribution. Events can be very close to the sensor and yet may not be detected. Finally the third model takes into consideration the target's path through the sensing area. The travel paths (best and worst) can be used to evaluate the sensing capability of the sensing network.

2.2.1 The Coverage Problem in WSNs

The coverage problem is defined by Wu and Tseng (2007) as: if S be the set of sensors such that $S = \{S1, s2, \dots, Sn\}$, in a two dimensional (2D) area A, and if S_i represents the i^{th} sensor where i = 1, . . ., n, and sensor S_i has location coordinate (x_i, y_i) inside A and a sensing range of r_i, then sensor S_i with a sensing range of r_i located at (x_i, y_i) can monitor points within a distance of r_i from itself S_i.

A point in the area *A* is covered by sensor S_i if it is within S_i's sensing range (r_i). A point in *A* is covered by *j* sensors if it is within *j* sensors' sensing ranges. The term subregion in *A* is a set of locations or points that are covered exactly by the same set of sensors. The coverage problem is defined as:

Definition 1: With a natural number *K*, the K-non-unit-disk coverage (K-NC) problem is how to determine whether all points in *A* are K-covered or not.

Definition 2: With a natural number K-unit-disk coverage (K-UC) problem is how to determine whether all points in *A* are K-covered or not, subject to the constraint that $r_1 = r_2 = \dots = r_n$ (Wu and Tseng 2007).

It is sometimes required that a WSN is k-covered such that $k > 1$. This is good for application where fault tolerance is needed, such as military surveillance. Other applications, such as triangulation, require a point in the sensing field to be at least 3-covered in order to get the desired results (Wu and Tseng 2007).

In Wang et al. (2003) a solution is proposed to address the K-UC issue. The coverage level is determined by looking at the coverage of intersecting points between sensors' sensing range. This paper claims that a sensing field *A* is K-covered if the intersecting points between each pair of sensors and between each individual sensor and the boundary of the sensing field *A* are at least K-covered. Using this property, a coverage configuration protocol (CCP) that schedules sensors on on-duty time basis while maintaining a coverage level for a given area is proposed (Wang et al. 2003). An area *A* has all its sensors actively monitoring the area. When this happens there will be areas that are covered by many sensors and as such some nodes will be redundant and go to sleep. A sleeping node is a sensor that has all its intersection points within its sensing range being at least K-covered by other nodes than itself in the neighbourhood. A sleeping sensor wakes up periodically to listen-state and evaluate whether to return to sleep-state or stay active depending on whether its sensing region is covered or not based on the previous discussion.

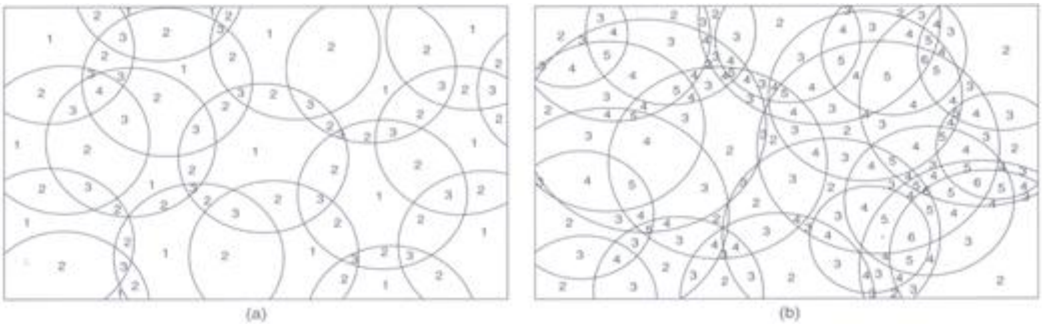


Figure 2. 1 Example of the coverage problem: (a) a sensing field which is 1-covered by unit disks and (b) a sensing field which is 2-covered by non-unit disks. The number in each sub-region is its coverage level (Wang et al. 2003).

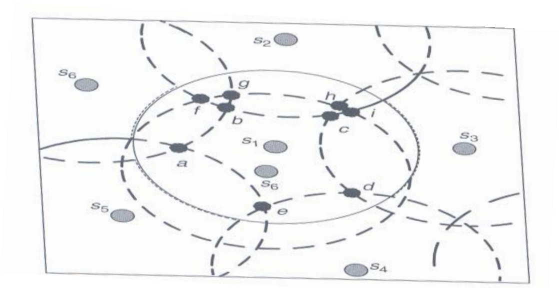


Figure 2. 2 A WSN example where sensor s_1 locally decides itself to be redundant and hence goes to sleep based on the approach proposed (Wang et al. 2003).

In Figure 2.2 the main objective is to have at least 1-coverage level in this sensing field. In order to determine whether sensor s_1 is eligible for sleep or not, sensor s_1 needs to inspect the intersection points, *a, b, . . ., i* within its sensing region. It could see that each point is covered by at least one other sensor, so it can switch to the sleep state otherwise it remains active. The two sensors that intersect at a point say ‘a’ are not considered during the coverage evaluation of the intersection point a. There is the need for coordinating the sensors from sleep to active state. In the example given above when sensor s_1 goes to sleep s_6 must be active hence the need for the coordination (Wang et al. 2003).

2.2.2 Coverage solution based on the Probabilistic Model

The binary model discussed previously is more suited for applications where sensors are required to sense temperature, humidity or light as sensing is only slightly affected by distance in these applications. However, in some applications, such as acoustic or seismic sensing, the sensing capabilities of the sensors are affected by distance, environmental factors and signal propagation characteristics.

The probabilistic sensing model is more suited for real world applications as it uses probability distributions to express the quality of surveillance of certain events sensed by sensors. This tries to take into consideration the environmental and the signal propagation characteristics along with the sensing capabilities of a sensor node. Using this model one can get the sensing capability of a sensor i for a location U expressed as a probability function $P_i(u)$ (Wang et al. 2003).

Let us say there are n sensors in a WSN and an event occurs at a location U, then the detection probability p(u) for these sensors can be modelled by:

$$P(u) = 1 - \prod_{i=1}^{n}(1 - p_i(u)) \qquad (2.1)$$

where, p_i is the sensing capability and depends on the signal propagation model.

The evaluation of sensing capability depends on propagation models. There are two well-known probabilistic sensing models for P_i which are briefly discussed below (Wang et al. 2003).

Signal decay model: n sensors are deployed at locations $l_i, i = 1, 2, \ldots, n$. The energy (Signal Strength) sensed by sensor s_i from a target emitting a signal at location U is given by,

$$E_i(u) = \frac{K}{||u-l_i||^{\mathfrak{R}}} + N_i \tag{2.2}$$

Where N_i is the noise, K is the energy emitted by the target and $\mathfrak{R}$ (typical from 2 to 5) is the signal decaying coefficient. $||u\text{-}l_i||$ denotes geometric distance between the target and the sensor s_i.

From this the sensing capability of sensor s_i for target u is defined as:

$$p_{i(U)=prob[E_{i(U)} \geq \eta\,]} \tag{2.3}$$

Where η is termed the receiver sensitivity, which is the minimum signal strength for a successful detection.

Log-normal Shadowing model: This is used to express the signal strength decay effect in an open area. PL is a path loss function used to express the decay effect in terms of d the propagation distance.

$$PL(d) = PL(d_0) + 10nlog\left(\frac{d}{d_0}\right) + X_\sigma \tag{2.4}$$

Where, n is the path loss exponent indicating the decreasing rate of the signal strength in that environment.

d_0 is the reference distance close to the transmitter

X_σ is a zero-mean Gaussian distribution random variable (in db) with a variance σ expressing the shadow effect.

From this the received energy of sensor s_i is derived as:

$$E_i(u) = K - PL(||u - l_i||) \tag{2.5}$$

2.2.3 Probabilistic Coverage Algorithm (PCA), an algorithm based on the coverage probabilistic model.

There are various protocols proposed for static sensor networks that schedule minimum number of sensor nodes out of the densely deployed nodes for efficient network coverage at any time. This is reliable as long as no coverage holes are created in the network due to nodes being turned off for energy savings. Protocols that assume single coverage includes Probabilistic Coverage in Wireless Sensor Networks (Di and Georganas 2002; Yan et al. 2003; Zhang and Hou 2003; Jiang and Dou 2004; Kumar et al. 2004; Ahmed al. 2005). In Wu and Tseng (2007) multiple coverage requirements are considered.

Probabilistic Coverage Algorithm (PCA) is an algorithm based on the coverage probabilistic model (Ahmed et al. 2005). Sensing is an important criterion of a wireless sensor network. However, the sensing coverage of a sensor node is usually assumed uniform in all directions (represented by unit disc), following the binary detection model. This assumption is unrealistic. In real world deployment the sensing capabilities of sensors are affected by environmental factors, therefore, it is imperative to factor this behaviour in coverage algorithms. The PCA algorithm explores the problem of determining the coverage, in a non-deterministic deployment of sensors using the technique of realistic probabilistic coverage model. This model captures the real world sensing characteristics of sensor nodes. This model is based on the assumption that the signal a sensor gets from a target follows a probabilistic model. This is best suited for applications such as object tracking and intrusion detection where a certain degree of confidence is needed in the detection probability. In such applications the signal strength fades away with distance from the source unlike the applications where temperature or humidity or light is measured. The PCA is based on the path loss log normal shadowing model and it is an extension of the Perimeter Coverage Algorithm (Ahmed al. 2005; Wu and Tseng 2007). This algorithm makes many assumptions and also requires the sensor to carry out significant computations. Hence in this algorithm the binary model approach has been selected and makes an assumption that the nodes are heavily deployed so that the area can be deemed to be properly covered.

2.2.4 Coverage Solution Based on Exposure

As a target moves further away from a sensor it becomes more difficult for the sensor to sense it. The sensor's sensing capability decreases as the distance from the target increases. Therefore in WSN applications where a target moves through the sensing field, a different coverage approach is assumed.

One method is indicated in (Wu and Tseng 2007; Meguerdichian, Koushanfar et al. 2001) whereby a path is located within the WSN which is the best or worst monitored as a target moves along the path. With this there is the *maximal breach path* and the *maximal support path* in a way that any point along such paths to the sensors is maximised and minimised, respectively. To find such paths polynomial-time algorithms are proposed.

2.3 Review of Routing Protocols

Routing protocols for WSN are predominantly multi-hop, that is as data are sent from source to destination many nodes are traversed before reaching the destination. In WSNs the destination of packets is mostly the *Base Station* (BS) or the *Sink*. Most of the time the base station connects the WSN to the outside world such as the Internet so that the users can get access to the collected data. Finding and Maintaining routes in WSNs are not an easy task because of the energy constraints and frequent unpredictable topological changes as a result of node failures. The main objective of routing protocols for WSNs is to maximise the WSN's life time.

In WSN, data quality is considered less important than energy conservation. Routing protocols can be categorised into two groups as shown in Figure 2.3.

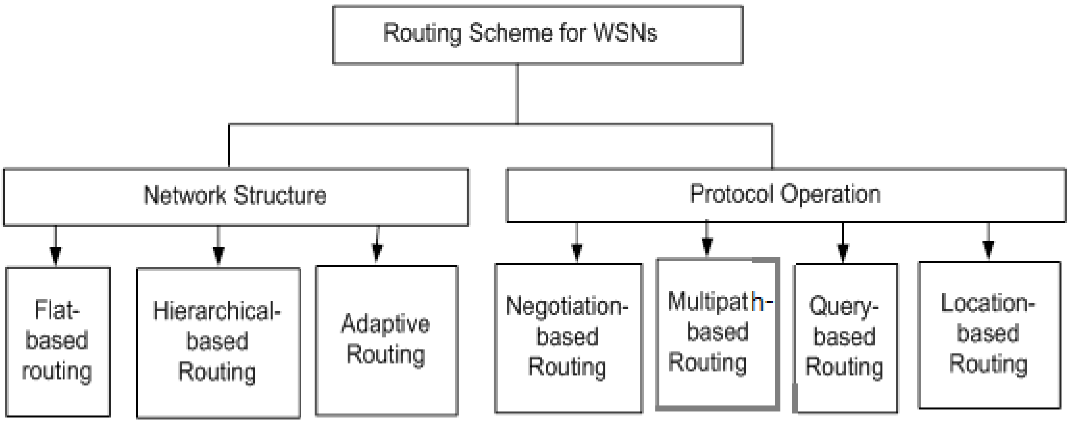


Figure 2. 3 Classification of routing protocols for WSNs

Each group can be further classified into smaller groups. All nodes are assigned equal roles in flat based routing while in hierarchical-based routing, nodes are assigned different roles and the Cluster Heads (CH) are given more roles to play. In adaptive routing the system parameters are controlled to adapt to the network conditions and energy availability in the nodes. The protocol operation group can be further divided into query-based, multipath-based, negotiation-based, or location-based routing techniques (Cordeiro and Agrawal 2006). In query-based routing queries, which are described in natural language or in high-level query languages, are used to get information within the network. In multipath-based routing, the routing protocols maintain multiple, instead of a single, paths to a destination. This increases resilience and load balancing within a network, thereby increasing network performance. A destination node sends a query, for a certain data (sensing task) from a node throughout the network. Any node that has the data matching the query will send it to the node that originated the query. In negotiation-based routing high level data descriptors are used so as to eliminate redundant data transmissions. The available resources are also used to dictate the communication decisions. In Location-based routing, the location of the data is as important as the data itself; hence the sensor nodes are addressed by means of their locations. The most common routing protocols will now be discussed.

2.3.1 Minimum Cost Forwarding Algorithm (MCFA)

Minimum Cost Forwarding Algorithm (MCFA) is based on the fact that the destination of a packet is already known and it is the Base Station (BS). Hence the routing is always done toward the BS. A sensor node (SN) does not need to have a unique Id or to maintain a routing table. The routing direction is already known, and this reduces the number of packets transmitted within the network and thus maximises the network life time. A node just needs to maintain a least cost path from itself to the BS. Messages that are forwarded by a SN are also broadcast to its neighbours by the SN. A receiving node always checks whether it is along the least cost path between the source SN and BS. If it is, then it rebroadcasts the message to its neighbours. This is repeated until the message gets to the BS.

The process through which each sensor node acquires the knowledge of the least cost path between itself and the base station is as follows. At the initial stage each node sets its least cost path to the base station to infinity. The BS then broadcasts a message to all the nodes in the network with the cost set to zero. A receiving node verifies whether the estimate in the message plus the link cost on which the message was received is less than the current estimate. If it is true, the current estimate and the estimate in the broadcast message are updated accordingly and the message is then re-broadcast, otherwise nothing happens. There is a potential problem here in that nodes farther away from the BS will get more updates and some nodes may have multiple updates. The MCFA is modified with the back off algorithm to solve this problem in (Jiang et al. 1998;Ye et al. 2001; Cordeiro and Agrawal 2006).

MCFA always selects the shortest route to the destination and does not take into consideration the power level of the nodes across such a route. This might result in draining the energy of some of the nodes completely thereby isolating the network, which is not desirable. Therefore, this method of routing will not be used in the proposed set of protocols.

2.3.2 Cluster Based Routing Protocol (CBRP)

Jiang et al. (1998) proposed Cluster Based Routing Protocol (CBRP) which is a type of hierarchical routing protocol (Jiang et al. 1998). In this approach the network is divided into clusters which are either overlapping or disjoint with two-hop-diameter in a distributed way. The CH aggregates the data from its sensor nodes and routes them to the destination. However, this comes with lot of hello packets for forming and maintaining clusters and may drain the energy of the network, which is undesirable (Jiang et al. 1998; Cordeiro and Agrawal 2006).

2.3.3 The Minimum Total Transmission Power-Routing (MTPR) algorithm

In Minimum Total Transmission Power-Routing (MTPR) algorithm (Scott and Bambos 1996), the best routes are selected based on the minimum total energy used in transmission along the route. There are bound to be problems as the energy level of the nodes along such path is not taken into consideration. This may result in draining some nodes. With this algorithm for a route l, of length D, the overall energy consumed over the route P_l is given by the equation.

$$P_l = \sum_{i=0}^{D-1} P(n_i, n_{i+1}) \tag{2.6}$$

Where, n_o and n_D $a$re the source and the destination nodes respectively. P (n_o, n_D) is the transmission power between two nodes. The selected route satisfies this property (Vergados et al. 2008).

$$P_k = P_l : l \in A\} \tag{2.7}$$

Where A is the set of all routes possible (Vergados et al. 2008).

2.3.4 Energy Prediction and Ant Colony Optimization Routing (EPACOR)

Shen et al. (2008) propose the Energy Prediction and Ant Colony Optimization Routing (EPACOR). In this routing algorithm when a node needs to deliver data to the sink, ant colony systems are used to establish the route with optimal or sub-optimal power consumption, and meanwhile, learning mechanism is embedded to predict the energy consumption of neighbouring nodes when the node chooses a neighbouring node to be added to the route.

In EPACOR, a mechanism is used to find the route with maximum remaining energy to the sink (remaining energy of a route is defined as the *minimum remaining energy of all the nodes in the route*). The nodes visited by an ant are recorded to avoid loops in building up a route to the sink. However the maximum remaining energy of nodes for a particular route might include some nodes with minimal energy thus depleting the energy of those nodes resulting in the partition of the network (Dimitrios, Nikolaos et al. 2008).

2.3.5 Random Walk Routing for Wireless Sensor Networks

In Tian et al. (2005) Random Walk Routing for Wireless Sensor Networks is proposed, a routing protocol based on random walk for a square grid topology is proposed. The proposed routing algorithm does not require any global location information and achieves load balancing for the WSN. The probability of successful transmissions from the source to the destination by random walk was analysed statistically and proved to be as energy efficient as the shortest path routing algorithm with the assumption that the message to be sent is small in size compared to the inquiry message among the neighbour nodes.

The algorithm provides load balancing in the WSN but the nodes near the base station inevitably experience heavier load than the nodes further from the base station. To remedy this situation a density-aware deployment scheme was used to guarantee that the heavily-loaded nodes do not affect the network lifetime even when exhausted.

This random walk protocol is purely based on equal probability and hence it may not be as energy efficient if some biasing was used in selecting the next hop such as the one with the maximum energy.

2.3.6 Directional Rumor Routing in Wireless Sensor Networks

The use of random walk in WSNs has been extensively researched, in Shokrzadeh et al. (2007): Directional Rumor Routing in Wireless Sensor Networks is a routing algorithm based on the random walk of agents. The aim of this algorithm is to improve the latency and energy consumption of the traditional algorithm using propagation of query and event agents in straight lines, instead of using purely random walk paths. Directed rumour routing has two phases for calibration.

In the first phase each node sends Hello messages stating its position to its neighbours. The hello messages are used by the receiving nodes to record their neighbours and their positions. In the second phase each node detects if it is at the edge of the network. When a node senses an event, it creates a number of event agents and propagates them into the network along some linear paths forming a star-like propagation trajectories. These event agents are not allowed to pass the edge nodes. After this a node is randomly chosen as the sink node. The sink node creates some query agents for each fired event. Each agent contains the Id of the current node, the Id of the previous node (depicting the direction of events), location information of the source node, and a table containing the Ids of the events and distances to them. The hello messages generated can drain the WSN of its limited bandwidth and hence hello messages will not be used in the proposed routing Algorithm.

2.3.7 Directed Diffusion Based on Clustering and Inquiry (DDBCI)

Directed Diffusion Based on Clustering and Inquiry (DDBCI) is an improved version of IDD (Improved Directed Diffusion) that is based on clustering. DDBCI is divided into four stages: cluster formation protocol state, Interest diffusion, Data propagation and Reinforcement (Yu and Zhang 2010). The major difference between IDD and DDBCI is that in DDBCI when a cluster head (CH) node receives an Interest message, it diffuses it into the cluster only if the target area of the Interest message is in the cluster, unlike IDD which will always do even if the target area is not in the cluster. In order to avoid diffusion of Interest into a non-targeted cluster, a CH in DDBCI maintains a table of member information whereby it queries its members of Interests received. If no responses are received it meant that the target area is not in its cluster and therefore the Interest is not unnecessarily diffused into the cluster. After the successful propagation of the Interest to the source node, the target nodes that get the Interest send their data to the base station through paths consisting of CH nodes and border nodes. The base station is likely to receive the data from various paths; however, it reinforces a data transmission path using certain criteria.

The first phase of the DDBCI is the cluster formation protocol which is made up of two sub-phases namely initial phase and cluster formation phase. A node can exist in the following states namely: ordinary nodes (ORN), cluster head node (CHN), cluster member node (CMN), candidate cluster head node (CCHN), and finally border node (BN). All nodes start as ordinary nodes and they may transition into other states after the cluster formation protocol process (Yanrong and Cao 2007; Yu and Zhang 2010).

The cluster formation protocol in DDBCI is the same as in IDD and is based on energy threshold TH_e, which is propagated by the base station to its neighbours and through the entire network. A node that receives this broadcast message will inspect the message and compare its residual energy to the energy threshold THe, setting. If the THe is greater than the node residual energy the node does not change its state. On the other hand if the node residual energy is greater than THe, the node transits to the CCHN state, and declares its state to its neighbours. In the case of two or more neighbour nodes being in the CCHN state at the same time, the tie breaker is their residual energies and the neighbour with the highest residual energy will win and transit to the CHN state. The winner then broadcasts its CHN state to its neighbours. The CCHN node that lost the tie breaker will then transit to the ORN state. The ORN nodes that receive a cluster head node message will join the cluster and set their states to CMN.

The ORN nodes that receive two or more cluster head node messages at the same time will set their node state to BN.

In order to maintain the clusters a CHN keeps a member information table to store the information of each CMN in its cluster. When the cluster is fully formed the CMN nodes put their ID information and position into a message called mem_msg and send it to their CHN node. When a CHN node receives a mem_msg it will check it against its member information table. If the record is already in the table the message will be discarded otherwise it is added to the table.

After the cluster protocol formation, the next phase is called the Interest Diffusion phase. The Interest message is diffused into the network by the base station and is only forwarded to the cluster head nodes and border nodes. The Interest message contains network information like target area, task type, data rate and timestamps. An *Interest table* is *kept by each sensor node to record Interest*. The *cluster heads keep a local Interest table where they record Interest and Interest attributes*. The Interest table for each Interest includesdata rate, neighbour cluster head node and timestamps which are used to set up the data propagation gradients. The cluster member nodes also have a local Interest table where they record the cluster head node that the Interest was received from. DDBCI uses the local Interest tables stored in the sensor nodes to set up data propagation gradient. A cluster head that receives an Interest will check its Interest table to see if the Interest has already been received. If that Interest is already in its Interest table then it will discard the Interest so as to avoid propagation of redundant Interest. If it has not received a similar Interest before, the CHN will then update its local Interest table with the new Interest. It will then inquire the member information table to see if there is any item that matches the target Interest. If there is no match, then the data does not exist in its cluster; therefore the CHN will flood the Interest to its neighbour CHN nodes and BN nodes. On the other hand if there is a match, then the Interest is in the cluster so it will then inspect the member information table and then unicast the message to the particular CMN that holds the matching data to the Interest (Yu and Zhang 2010).

The next phase is the Data Propagation Phase. If the Interest propagated has a data match from a CMN node, BN node or a CHN node then there are two sets of propagation methods depending on the node type. The BN node or CMN node with a match of Interest collected and data sensed, will send the data to the neighbour CHN node first and then to neighbour CHN nodes in gradients directly. In the case that the data source node is a cluster head node then the data matching the Interest will be sent to neighbour CHN nodes in gradients directly. With this scenario a CHN node is likely to receive or send sensing data from many neighbour CHN nodes; therefore the base station may receive the same sensing data from different paths across the network. So, in order to avoid the unnecessary duplication of sensed data transmission to the base station a middle CHN node that receives a sensing data from other CHN searches its Interest table for matching Interest entry in its cache. *If there is no match then the message is discarded*. If there is a match, the data is also discarded as the match indicates that the sensing data has already been forwarded. The CHN checks information of the neighbour cluster head and based on the neighbour's transmission rate the node may send all the received data to the appropriate neighbours or send the data to the neighbour in proportion to their capacity (Yu and Zhang 2010).

The next phase is the Reinforcement phase. During the initial data propagation phase several transmission paths from the source node to the base station are set up. In order to establish the probe gradient to the source node, border nodes and cluster head nodes transmit the sensing data at a lower rate. After this initial low rate data transmission, the base station sets up a reinforcement path so as to get the best path from the source node to the base station. Once the best path is selected the next incoming data will be transmitted along this path at a greater rate (Yu and Zhang 2010). Figure 2.4 depicts an example of the operation of DDBCI.

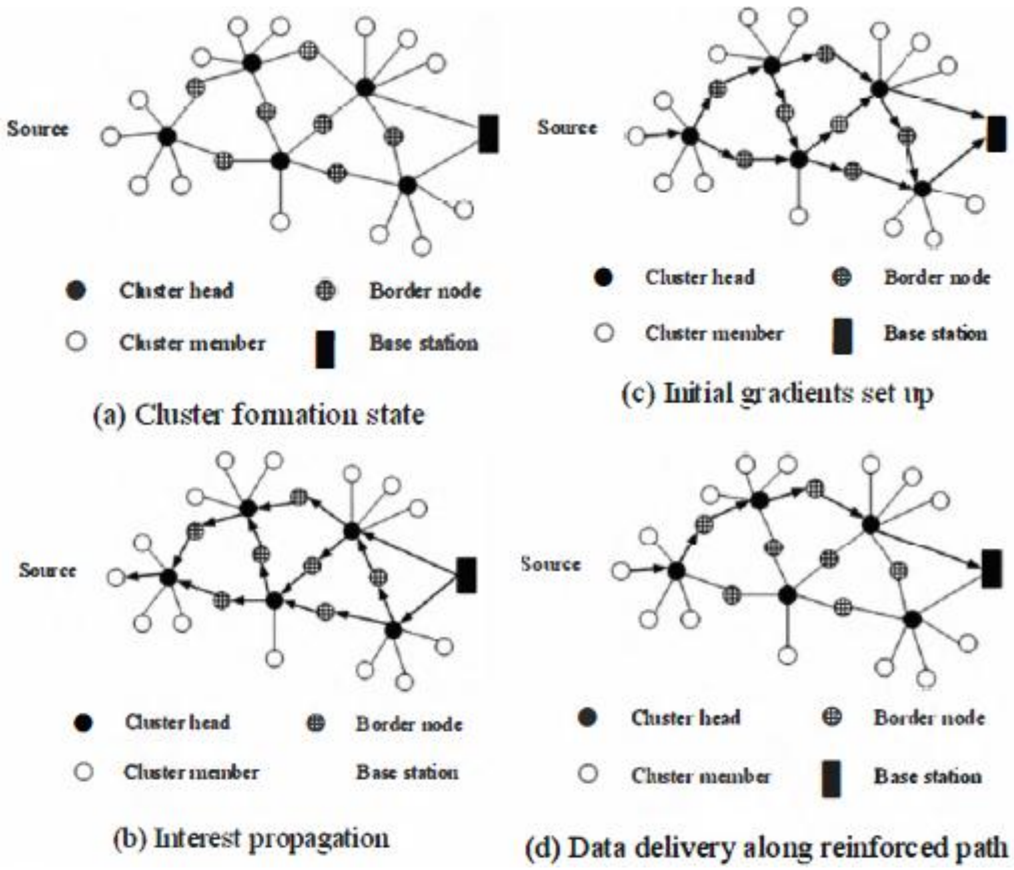


Figure 2. 4 A simplified schematic for DDBCI (Yu and Zhang 2010).

DDBCI was simulated using C++ programming language and compared with DD and IDD. In the simulation results DDBCI protocol has a higher energy efficient and smaller delay.

2.3.8 The LEACH routing protocol

Low Energy Adaptive Clustering Hierarchy (LEACH) is a clustering-based protocol that randomly rotates local cluster heads so as to load balance the energy load requirement among the sensors in the network. Its data aggregation technique reduces the amount of information to be sent to the base station thereby reducing the energy usage of the network as computation is much cheaper than communication (Heinzelman et al. 2000b).

With LEACH many clusters are created with each cluster having a cluster head whose main job is to collect and aggregate data from cluster member nodes and transmit the data to the base station directly. With a large network there will be many clusters and therefore some of the cluster heads far away from the base station will not be able to transmit directly to the base station. This is one of the limitations of *LEACH* which will be addressed in the proposed protocol *C-BRALB*.

2.3.9 The HEED routing protocol

Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks (HEED) is an iterative clustering protocol that is primarily based on the remaining energy of each node which is fairly estimated. Intra-cluster “communication cost” is considered as a secondary clustering parameter whereby cost can be a function of neighbour proximity or cluster density.

In HEED the primary clustering parameter is used to probabilistically choose an initial number of cluster heads whereas the secondary parameter is used to break a tie among cluster heads in the case that a node falls within the “range” of more than one cluster head. The cluster range or radius depends on the node configurable transmission power level used for intra-cluster announcements and during clustering. These cluster power level needs to at least cover two or more cluster diameters for the resulting inter-clustering overlay to be connected.

Intra-cluster communication cost which is the secondary clustering parameter depends on cluster properties (e.g. size) and the type of intra-cluster communication allowed. In the case that all the nodes use the same power level for intra-cluster communication then the cost can be set to be directly *proportional to node degree* with the objective of *load balancing the load among cluster* heads, or *inversely proportional* to the node degree *with the objective of creating dense clusters*.

In the case that different power levels are used for intra-cluster communication then the cost for a node to select a cluster head depends on the expected intra-cluster communication energy consumption rather than the nearest cluster head. HEED is a very good protocol but it comes with loads of overhead as a result of its sophisticated clustering technique which requires huge processing power for the sensor nodes (Younis and Fahmy 2004).

2.3.10 Proposed Protocols

There has been much research into *regular topologies* for WSNs, to efficiently save energy and hence extend the lifetime of the network. However, little work has been carried out on routing in patterned WSNs; except for the shortest path first (SPF) routing algorithms, which uses global location information. In this work initially a Biased Random Energy-Efficient Routing Algorithm has been proposed (BRALB). It is based on random walk and probability; hence it does not require any global information. It uses statistical estimation to acquire all the information it needs to route packets based on energy resources in each node without requiring any global information.

It is shown in this work that the random walk routing uses the same energy as the shortest path first routing in applications where the message to be sent is comparatively small in size, with the inquiry message among the neighbours. It is also shown to balance the load (i.e. the packets to be sent) among the neighbouring nodes. It is known that in many WSN applications the messages sent to the base station are very small in size: Therefore BRALB is viable and can be used in these applications. BRALB has been demonstrated to be a statistically and empirically effective WSN routing algorithm (Touray and Johnson 2012b).

Despite all the good features BRALB has scalability issues as the average packet delay increases as the network depth increases. To remedy the issue BRALB is modified with the introduction of a clustering scheme, resulting in a new routing algorithm called C-BRALB. In C-BRALB when a node has a datum to transmit it sends the data to its cluster head. It is the duty of the cluster head to aggregate traffic and transmit it to the base station through neighbouring cluster head. A cluster keeps a counter for each neighbour where it records the messages it sent to or received from them. When a cluster head (CH) wants to forward data it inspects its neighbour cluster counters and then forwards the data to the CH whose counter has the least message. The messages sent or received by a cluster can be used to estimate the energy value of a cluster. Therefore the CH can always route to the cluster with more energy by inspecting its CH counters and hence load balance the traffic across the network. The clustering scheme proposed is based on the IDD clustering technique, the same technique used by DDBCI, however, in C-BRALB some modifications are made to make the clustering technique more efficient. The major differences are the omission of Interest diffusion phase and the introduction of a total different Data Propagation phase. C-BRALB is composed of two phases which are the Cluster Formation and Data Propagation (Touray and Johnson 2013).

2.4 Summary

In this chapter, existing research work on routing algorithms and clustering algorithms for WSNs were reviewed. Research challenges and performance optimization techniques required to achieve energy efficient routing algorithms are also discussed. The various routing algorithms have also been compared in this chapter on the basis of routing overheads, the hierarchical architecture, scalability, energy efficiency and fault tolerance.

Chapter 3

WIRELESS SENSOR NETWORKS

Chapter 3

WIRELESS SENSOR NETWORKS

3.1 Introduction

A wireless Sensor Network (WSN) consists of a collection of minute nodes organised into a cooperative network (Hill et al. 2000). A WSN comprises of processing capability (microcontrollers one or more, CPUs, or DSP chips), may have multiple types of memory (program, data and flash memories), an RF transceiver (very often with a single Omni-directional antenna), and a power source (batteries or solar cells or both), and various actuators and sensors. These features of the sensor node are all limited in resources and hence various schemes and methods or algorithm have been devised to maximise their efficiency. The nodes are most of the time deployed in an ad hoc fashion, communicate wirelessly, and are often self-organised (Stankovic 2006). These functional components of sensor nodes are shown in Figure 3.1 (Biswas et al. 2005).

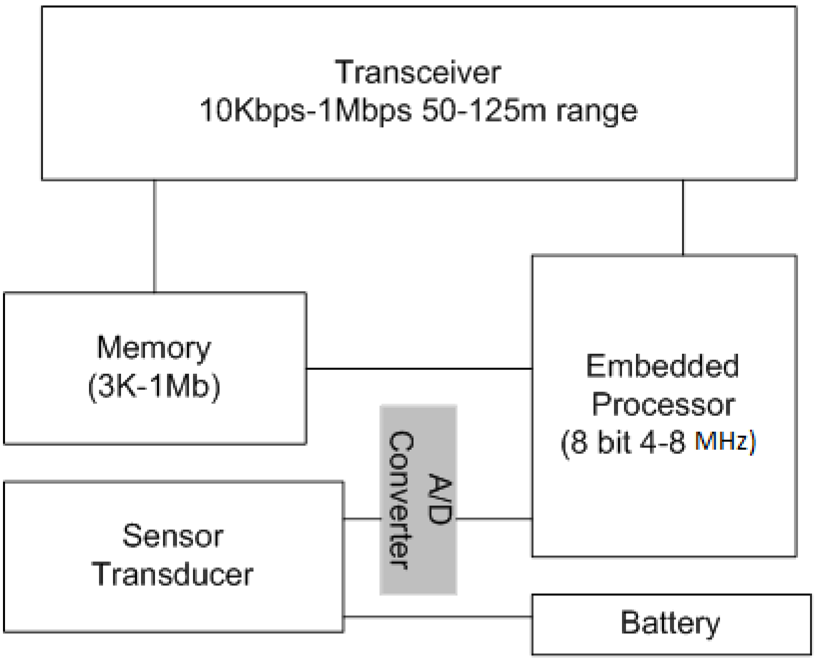


Figure 3. 1 Functional Block Diagram of a typical Sensor Node (Biswas et al. 2005)

The sensor transducer is used to sense a given physical quantity with a degree of predefined precision, an embedded processor for local processing, a small memory unit for storage of data, and a wireless transceiver to receive or transmit data. All the power needed by the node for these activities is supplied by the battery.

WSNs can be employed for different applications including among others industrial, habitat and environmental monitoring; these have great importance for mankind as a whole. One of the most critical issues in the wireless sensor networks is the limited availability of energy in the nodes. The network lifetime strictly depends on its energy efficiency. Some of the schemes used for extending and maximising the network life time are energy-aware technologies, in-network processing, network topology, multi-hop communication, and density control techniques. WSNs should be fault tolerant due to various reasons such as dying of nodes due to energy depletion or physical destruction of nodes. The mechanisms for fault tolerance can take good advantage of node redundancy and distributed task processing as the nodes are deployed in large numbers (Boukerche 2008)

As there are various applications for sensors there are also different types of sensor nodes with different capabilities and designs to suit each application. Some sensor nodes use small batteries which last for a day or two while others have a larger battery capacity and can last for a month with continuous operation. For some applications a renewable power supply such as solar panel is used. Some are even embedded into other devices and as a result get their required energy from them. In such scenarios sensors are not limited in energy and can therefore use a wireless single-hop access to infrastructure networks such as the Internet to send the data collected.

For other applications, sensors can be of large size protected in boxes or placed at a height so as to improve their communication and protection from damages. For applications such as seismological data or acoustic bird presence detection the data is not time critical and hence the sensor network may not be attached to infrastructure networks such as the Internet (Nayak and Stojmenovic 2010).

Another important aspect of WSNs is the mode of deployment. Depending upon the application requirements, various architectures and node deployment schemes have been proposed and developed for WSNs. The method of node deployment can be either random or deterministic. These also depend on applications and place of deployment. Whether deterministic or randomised method of deployment is used, the main issue is the coverage of the targeted areas of interest (Gajbhiye and Mahajan 2008).

Node deployment is the basic problem in WSNs and it depends on the application. If the environment to deploy the nodes is known and under control, then the node deployment can be a lot easier. However, the reality is that in some applications the environment can be unknown or hostile or both, such as enemy territory in battle fields, disaster areas and toxic urban regions and harsh fields, in such situations sensor deployment is not performed manually. One solution is the deployment of the sensors by aircraft scattering. The issue with this technique is that there is no control on the actual landing position of the sensors due to wind, trees or buildings. These may result in an inferior coverage as required by the application irrespective of the number of nodes deployed. In toxic-leak detection mobile sensors can be deployed within a safe area from the toxic area and then they move to the correct positions to provide the required coverage so as to be able to gather the required information (Howard et al. 2002a; Howard et al. 2002b) .

For static environments, deterministic deployment can be used since the location of the sensors can be predetermined properly. BRALB, the protocol proposed in this work, is meant for such environments. When the information of a sensing area is not known in advance or changes with time, then stochastic deployment can be used, as the position for sensor deployment cannot be determined (Huang and Tseng 2003; Meguerdichian et al. 2001).

Coverage is an important aspect for WSNs. The main goal of a WSN is to monitor some physical quantity in a given area; therefore there is the need for adequate density of sensors for the target area in order to avoid black holes in the area due to uncovered sensing regions. In order to achieve a complete sensing of the area two important factors need to be considered namely the sensing range of each sensor and the area to be covered.

The sensors can be placed randomly or deterministically at pre-specified locations. Let N be the number of sensors in a given square grid area: A = L*L, where L is the length of the grid, therefore the *density of the sensors* is given by $\lambda = N/A$. Let the sensing range for a sensor is r_s. For the sensor to cover the whole space, adjacent SNs must be located close to each other and at most at a distance of $2r_s$ from each other as shown by the positions of SN_1 and SN_2 with varying overlapping areas between SN_1 and SN_2.

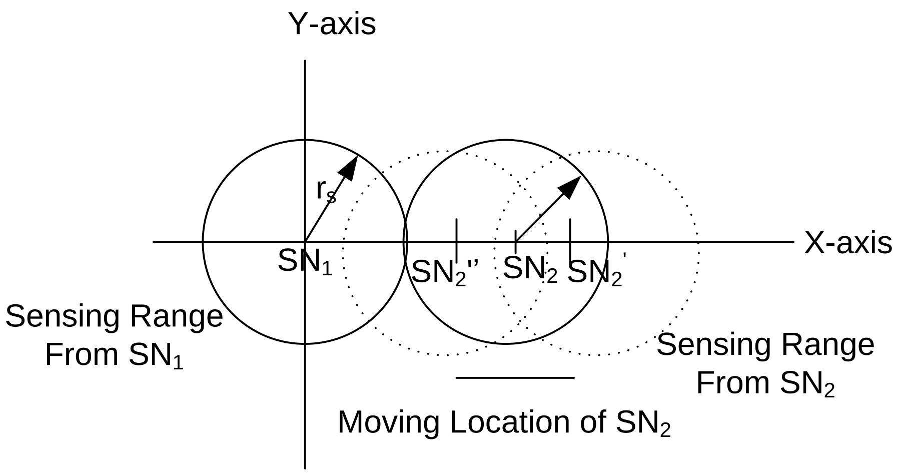


Figure 3. 2 Sensing Range of a SN

The relationship of a number of sensors uniformly distributed with a node density of λ and the probability that there are m sensors within the space of S is Poisson distributed as given by (Christian 2002) :

$$P(m) = \frac{(AS)^m e^{-\lambda s}}{m!} \quad (3.1)$$

Where, space $S = \pi r_s^2$ for two dimensional spaces. Equation 3.1 gives the probability that the monitored space is not covered by any sensor and therefore the probability P_{cover} of the coverage by at least one SN is:

$$p_{cover} = 1 - P(0) = 1 - e^{-\lambda s} \tag{3.2}$$

By following this relation one can have an idea about the number of sensors required for an adequate coverage of a given area.

Fully sensing the targeted area is very important but it is meaningless if the sensed data cannot be transmitted. Hence, not only the sensing coverage, but also the communication connectivity is equally important so that the sensed data could be transmitted across the network to the sink for collection and analysis. The communication range of each sensor can be deduced from the sensing coverage of the sensors.

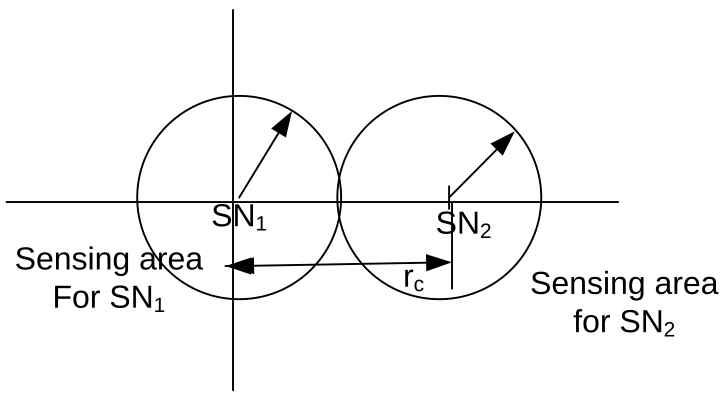


Figure 3. 3 Sensing and Communication range of SNs

In Figure 3.3 two sensors can be separated by a maximum distance of $2r_s$ which is equal to r_c for full sensing coverage (Boukerche 2008). The communication range of a node, also known as the transmission range is r_c, and the maximum distance between say SN_1 and SN_2 is $2r_s$. Therefore, in order to transfer data between SN_1 and SN_2 the maximum transmission range must be $2r_s$. Hence the *wireless communication coverage of a sensor must be at least twice the sensing distance* and it is the *minimum* for connectivity between sensing devices to communicate with each other (ZHANG 2005) .

Another important thing next to the sensing and communication coverage is how those sensed data get transmitted across the WSN. This leads to the network layer and the protocols responsible for the sending of data across the WSN. Some of the classes of protocols for WSNs are briefly introduced below:

- Localised algorithms describe the class of protocols in which a node makes its routing decision based *on local and limited information* instead of a global information about the WSN (Boukerche 2008).
- Centralised protocols are executed on a central node usually using global information of the WSN. With the high number of nodes in WSNs it is very unusual to use such protocols, as the cost of acquiring global network information is most of the time unfeasible in most WSNs.
- Distributed algorithms are related to different computational models. The computational model in a WSN is a set of sensor nodes that communicate among themselves using a message passing mechanism (Boukerche 2008).

Some of these protocols operate at the MAC layer, while others operate at the application or network layer. *For this research the network layer protocols are the main subject*. The most important protocols for the network layer are the routing protocols. Routing is the process that enables sending of messages in the form of data packets from source to the destination. The way of transmitting the message can be in the form of broadcasting whereby the message is sent to all the neighbour nodes irrespective of whether they need it or not. It could also be in the form of multicast whereby the message is sent to a target group of sensor nodes or it could be unicast where it is sent to a target sensor node. It could be a combination of the above mentioned schemes for transmitting messages. All these methods have their advantages and disadvantages and the method selected greatly depends on the application considered. The routing protocols can be classified into three types: flat-based routing, hierarchical-based routing and adaptive-based routing. These will be discussed in the following sections in detail.

3.2 Sources of Communication Inefficiency

Reducing the energy consumption of WSNs is one of the critical design challenges of WSNs. The major components that consume energy in WSNs are: sensing unit (Sensing transducer and A/D converter), communication unit (transmission and receiver radio) and the computing/processing unit (Cordeiro and Agrawal 2006). The non-ideal behaviours of a real wireless system lead to the actual performance of communication to differ substantially from idealised assumptions. These inefficiencies arise from a number of sources at various levels of the system hierarchy. Some of these inefficiencies are shown in Figure 3.4.

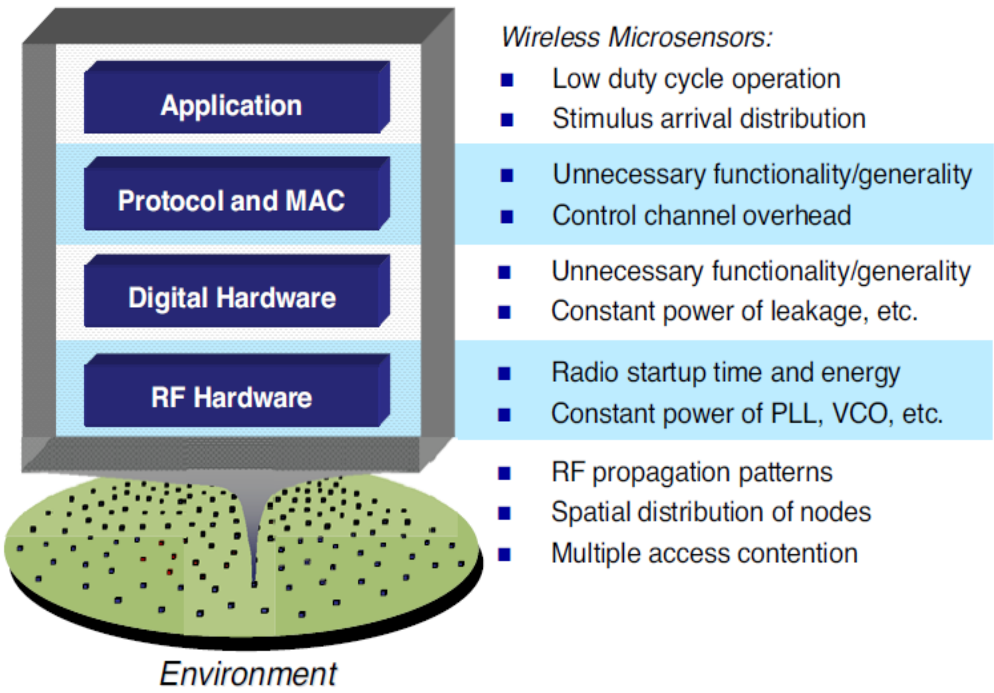


Figure 3. 4 Sources of inefficiency in communication across the system hierarchy (Min 2003).

3.2.1 Radio Transceiver Energy

Two key issues are the static power and the cost of shutdown. The radiated RF (radio frequency) energy and static power dissipated by the analogue electronics are the energy consumed by the radio transceiver. The radiated energy is proportional to the transmitted distance as d^2 to d^4 depending on the condition of the environment. This radiated energy has historically dominated the other radio energy factors. However, for closely packed micro sensors the radio electronics are of even greater concern.

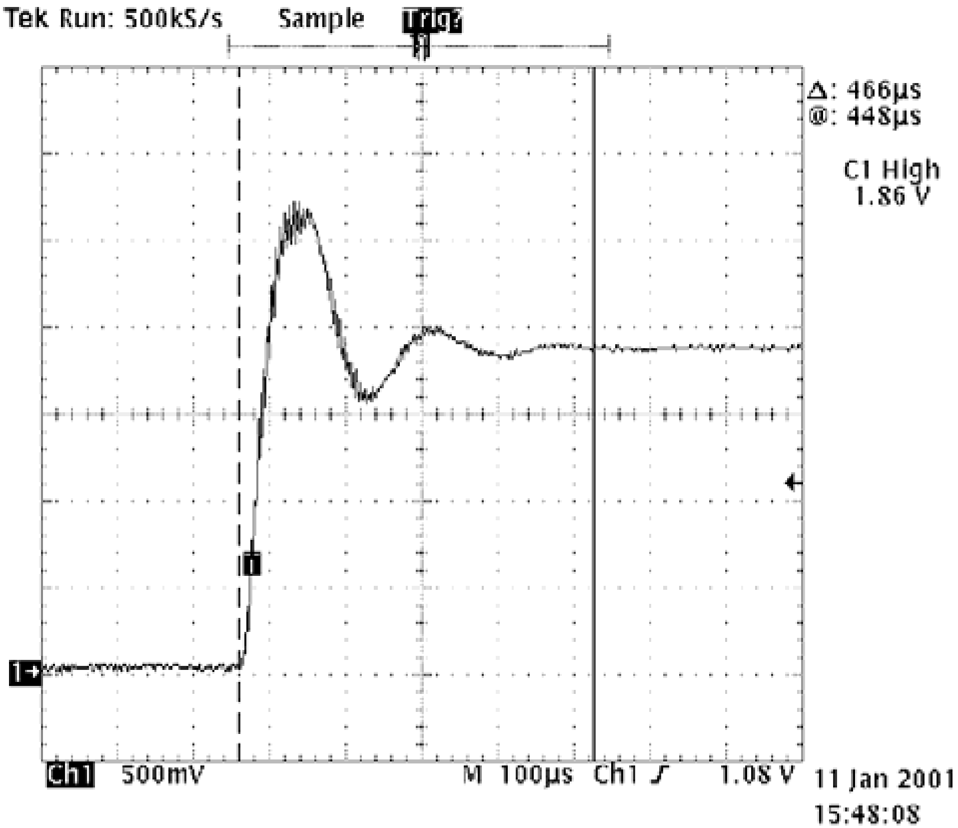


Figure 3. 5 466 µs startup transient from the µAMPS-1 2.4 GHz radio, measured at the input to the radio's VCO (Min 2003).

The average power consumption of a micro sensor radio is given by:

$$p_{radio} = N_{tx}[p_{tx}(T_{on-tx} + T_{st}) + p_{out}T_{on-tx}] + N_{rx}[p_{rx}(T_{on-tx} + T_{st})] \qquad (3.3)$$

Where N_{tx}/N_{rx} is the average number of times per second that the transmitter is used, P_{tx}/P_{rx} is the power consumption of the transmitter/receiver in watts, P_{out} is the power output per transmission, T_{on-tx}/T_{on-rx} is transmit/receive on time, (actual data transmission/reception time) and start-up time of the transceiver is T_{st}. The ratio of the packet size L in bits to R the radio data rate in bits per second is $T_{on-tx/rx}$. The amplifier power is considered to be 'on' only during actual data communications.

During start-up time T_{st} there is no data transmission as the internal phase-locked loop (PLL) of the transceiver must be locked to the frequency of the desired carrier frequency before the successful demodulation of data (Min 2003). This start-up time is a very crucial factor and Figure 3.5 shows the graph of the start-up transient of a commercial 2.4 GHZ transceiver (National Semiconductor Corporation 1999). The graph shows the control input to the voltage-controlled Oscillator (in volts) versus time in Figure 3.5.

For applications whereby the sensor nodes tend to communicate with very short packets there is a significant impact of the start-up time on the average energy consumption per bit E_b. The turning off of radio during idle period can have serious ramifications as large amount of energy is dissipated during initial start-up time of hundreds of microseconds. The effect of start-up transient is shown in Figure 3.6 for Ptx=220mW.

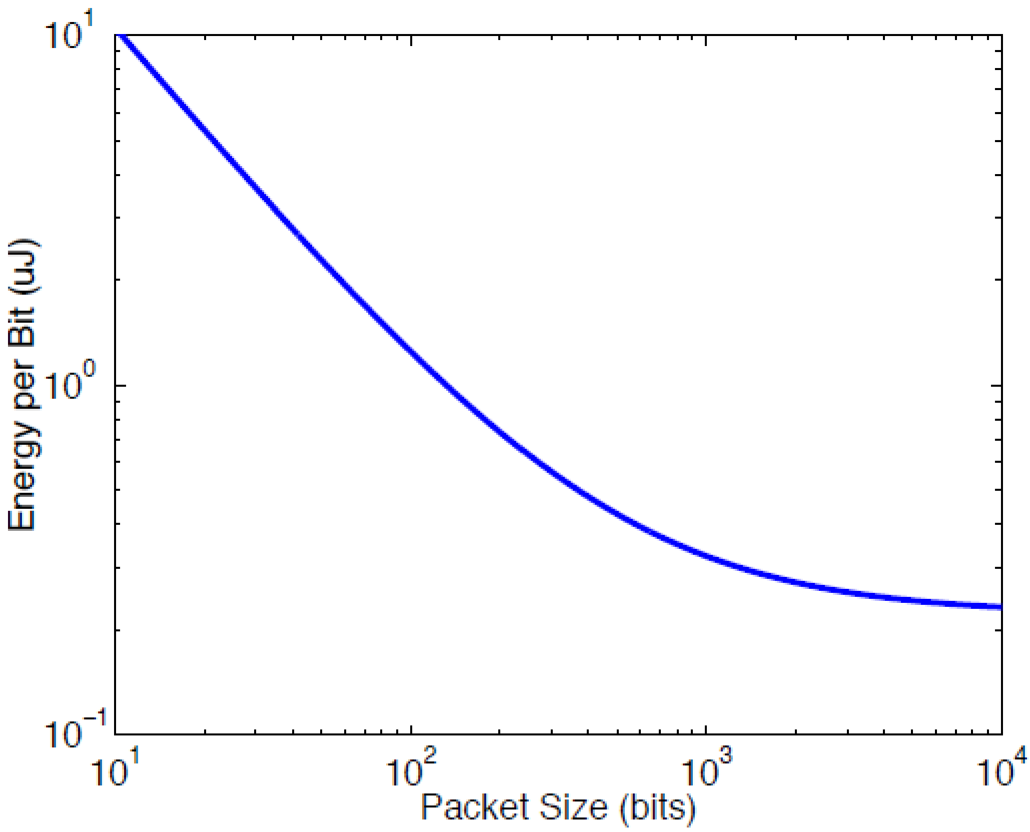


Figure 3. 6 The total radio energy per bit required to transmit packets of various sizes. The radio has a data rate of 1 Mbps, start-up time 466 µs, and the active transmitter electronics consume 220 mW (Min 2003).

The graph shows energy consumption per bit versus packet size. This shows high energy cost per bit for reduced packet sizes, which is the effect of the start-up time energy consumption. For large packet sizes, the fixed energy cost of radio start-up is spread over more bits resulting in less average energy per transmitted bit while for short packet sizes the opposite is true.

3.2.2 Environmental Inefficiency

The communication and sensing of a WSN node is affected by the environment in which it operates. With random deployment of sensor nodes it is very certain that the nodes will not be uniformly spaced and the node density might vary from low to high, all these irregularities could increase the communication burden on the individual nodes. The spatial variation of the environment could also bring similar issues. These spatial variations in the environment could be in the form of obstacles, varying temperatures, or pressure or light intensity, uneven terrain or Radio Frequency Interference (RF). All these environmental factors could cause a variation of the communication channel with location.

3.2.3 Digital processing Energy

The energy consumption by digital processing unit is mainly of two parts: the switching energy and the leakage energy. The switching energy is independent of time while the leakage energy is linear with time. These two parts sum up to give the total energy consumed by the digital circuit as follows:

$$E_{digital} = cV_{DD}{}^{2} = tV_{DD}I_{0}e^{\frac{V_{DD}}{nVT}} \tag{3.4}$$

The switching energy term is represented by CV_{DD}^{2}, with V_{DD} representing the supply voltage and C the switched capacitance. The switching energy is the energy required to switch parasitic capacitors on an integrated circuit (IC). The undesirable leakage energy is as a result of current leakage from power to ground at all times. This value is set by the thermal voltage V_T and constants n and I_0 which are processor parameters and can be obtained from experiments.

Switching energy is the dominant part of digital processing energy, however, for modern CMOS applications the leakage energy is increasingly becoming dominant as a result of recent semiconductor process technologies (Weste and Eshraghian 1993. The leakage energy does not follow the Morse Law, however, for each new processor generation leakage energy increases by threefold. Leakage in advance technologies is approaching 50% of a digital circuit operating power (De and Borkar 1999).

Figure 3.7 below illustrates the relationship between the switching energy and the leakage energy as the operating frequency of the nodes CPU is varied.

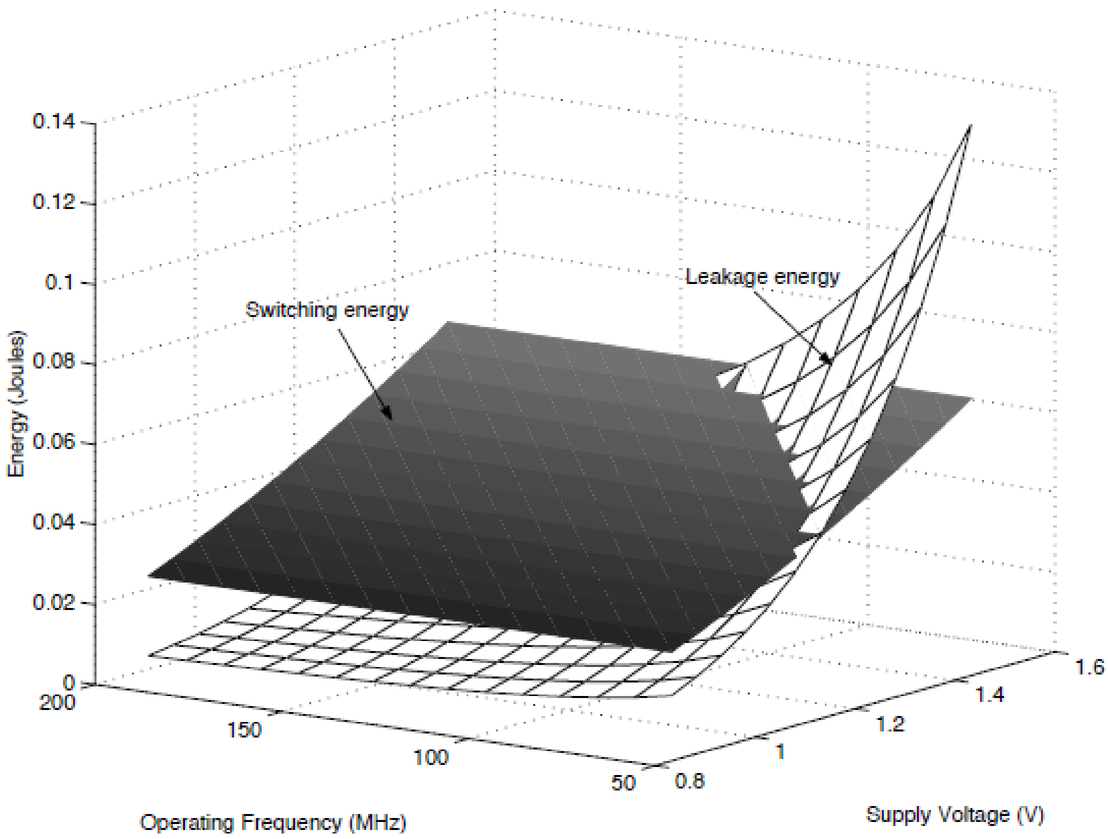


Figure 3. 7 Comparison of leakage and switching energy in SA-1100 (Sinha and Chandrakasan 2000).

The graph shows that the leakage energy begins to dominate as the processor's frequency is reduced. The leakage energy is linearly proportional to time irrespective of work being done or not. The simplest way to solve this is by powering off nodes which again requires time and energy overhead. A shutdown policy needs to consider the energy saved in shutting down against the hidden time and energy to shutting down the node.

3.2.4 Protocol Overhead

Media access and routing protocols have received a great deal of attention in recent years. These protocols create an additional overhead to wireless communication. In order to function properly these protocols require control packets which brings an additional cost. These control packets are needed to implement route discovery and maintenance, channel reservation, authentication and even somehow ironically power management. The 802.11b media access protocol specifies four data frame types and 21 types of management and control frames.

The energy overhead of a packet head can be very significant. The header can sometimes be several times larger than the actual data. In the TCP/IP protocol stack Telnet protocol would carry a 400% overload relative to the amount of data it actually carries (Nagle 1984.). This is the reason why in some traditional routing protocols some TCP compression techniques are employed. These compression techniques require large CPU usage which cannot be found in the limited CPU of a sensor node.

3.3 Average Power Dissipation of a Node

A lithium primary battery offers an energy density of almost 2 kilojoules per cm^3 (Hahn and Reichl 1999). With 2 KJ capacity the energy can transmit 600,000 ten-kilobits packets (nine minutes of active transmission) over the 802.11b commercial transceiver (Gruteser et al. 2001), the Viterbi decoding of 230,000 packets on some commercial low-power processors (Shih et al. 2001). These seem fairly reasonable from that perspective but the actual fact is that WSNs do not operate all the time in active mode and the most important criteria is not the number of packets it can actively transmit but how long the battery can stay and deliver packets is the most important part. If the network should be up for a whole month the average power dissipation must be less than (2000J)(1day/24hrs)(1hr/3600s) = 23.1mW and for one year the average power dissipation must be less than 63.4μW. The standby power dissipation of most sensor nodes exceeds 63.4μW. Therefore energy efficiency is of paramount concern in WSNs. The energy density of batteries does not follow Moore's Law, but only doubles every five to twenty years (Powers, April 1994).

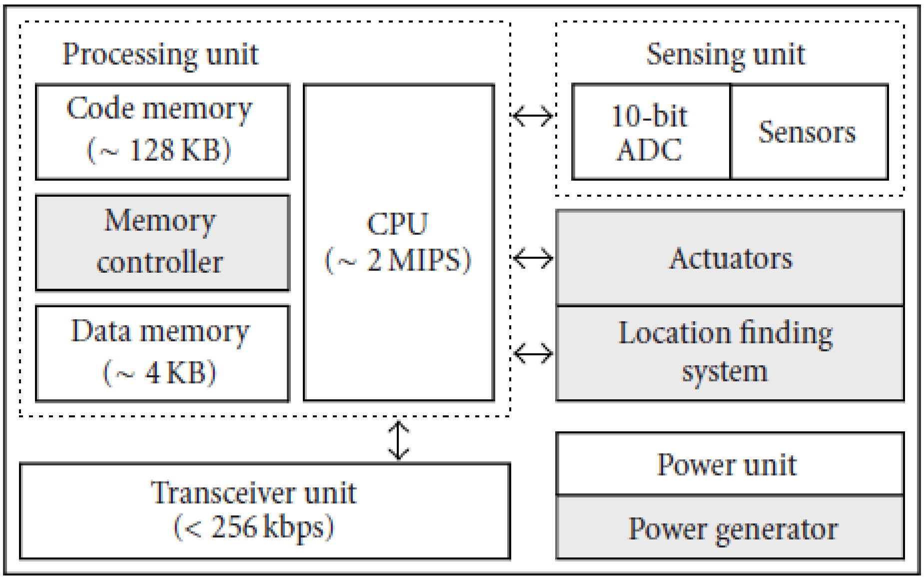


Figure 3. 8 Reference hardware platform architecture of a sensor node (Kuorilehto et al. 2005).

The values given in Figure 3.8 are the resources available in MICA2 mote (Crossbow 2012). The power consumption of a node varies from one design specification to another, but it is normally in the order of mW when active and in the order of μW when in sleep mode. Their power unit is a typical AA battery or a similar energy source.

Table 3.1 gives the parameters of different sensor nodes (Cordeiro and Agrawal 2006). The smallest power utilization during sleep mode is 0.001mW in a Telos 2004 node while the minimum energy used during active period is 60mW for generic sensors. With 60mW going by the calculation illustrated above the power would not even last for 3 months if it is only active and not even sending data. With these considerations, there is an urgent need for efficient usage of the limited energy resources in a wireless sensor node. These limitations are one of the issues addressed by this research work through the scope of a routing protocol.

Table 3. 1– Parameters of different Sensor Nodes (Jason Hill June 2004).

Spec 2003	4–8Mhz Custom 8-bit	3mW peak 3uW idle	3K RAM	I/O Pads on chip, ADC	50–100Kbps	Full custom silicon, traded RF range and accuracy for low-power operation.
Generic Sensor Nodes						
Rene 1999	ATMEL 8535	.036mW sleep 60mW active	512B RAM 8K Flash	Large expansion connector	10Kbps	Primary TinyOS development platform.
Mica-2 2001	ATMEGA 128	.036mW sleep 60mW active	4K RAM 128K Flash	Large expansion connector	76Kbps	Primary TinyOS development platform.
Telos 2004	Motorola HCS08	.001mW sleep 32mW active	4K RAM	USB and Ethernet	250Kbps	Supports IEEE 802.15.4 standard. Allows higher-layer Zigbee stardard. 1.8V operation
Mica-Z 2004	ATMEGA 128		4K RAM 128K Flash	Large expansion connector	250Kbps	Supports IEEE 802.15.4 standard. Allows higher-layer Zigbee stardard.
High-bandwidth Sensor Nodes						
BT Node 2001	ATMEL Mega 128L 7.328Mhz	50MW idle 285MW active	128KB Flash 4KB EEPROM 4KB SRAM	8-channel 10-bit A/D, 2 UARTS Expandable connectors	Bluetooth	Easy connectivity with cell phones. Supports TinyOS. Multihop using multiple radios/nodes.
Imote 1.0 2003	ARM 7TDMI 12-48MHz	1mW idle 120mW active	64KB SRAM 512KB Flash	UART, USB, GPIO, I^{2}C, SPI	Bluetooth 1.1	Multihop using scatternets, easy connections to PDAs, phones,TinyOS 1.0, 1.1.
Gateway Nodes						
Stargate 2003	Intel PXA255		64KNSRM	2 PCMICA/CF, com ports, Ethernet, USB	Serial connection to sensor network	Flexible I/O and small form factor power management.
Inrysnc Cerfcube 2003	Intel PXA255		32KB Flash 64KB SRAM	Single CF card, general-purpose I/O		Small form factor, robust industrial support, Linux and Windows CE support.
PC104 nodes	X86 processor		32KB Flash 64KB SRAM	PCI Bus		Embedded Linux or Windows support.

3.5 Single-hop Wireless Sensor Networks

In single-hop wireless sensor networks the sensed data is sent directly to the base station or sink without any intermediate nodes using the wireless medium. For such applications, the communication range of the individual sensors must be able to reach the sink in a single hop. The power required to send the data must also be supported by the battery. In such type of applications it is very common to embed sensors into a different device. In health care applications for example, sensors are embedded in watches and then attached to the patients. These sensors can then monitor blood pressure, pulse and temperature, and periodically send the data to the sink which can be a laptop in this case. The sensors can also be programmed to send an alarm to the sink when certain critical measurement thresholds are reached.

John et al. (John et al. 2005) uses sensors to analyse the condition of the rehabilitation of stroke patients. As the patients exercise, the sensors attached to them monitor their muscle and limb movements. The data obtained is then sent by a single hop to the control centre, the sink which is a laptop. The *system architecture* is shown in Figure 3.12 and was able to capture various set of motion data which was used to study the effect and progress of the rehabilitation exercise.

3.6 Multi-Hop Wireless Sensor Networks

In general, nodes in WSNs are densely and randomly deployed. The nodes may be monitoring a large area depending on the application. In such cases the nodes might not be able to transmit sensed data to the sink in a single hop. This is as a result of the limitations of the transmission range and energy of a sensor. Therefore, a more energy efficient way is for the sensors once they are deployed, to self-organise themselves into a wireless network and operate in multi-hop fashion. Various routing and clustering techniques have been proposed to address this need. In a WSN while implementing a clustering mechanism, it is the responsibility of the cluster head (CH) to collect sensed data from the nodes belonging to its cluster and sending it to the sink through other nodes along the path to the sink. For normal operation where no clustering is used a sensor node sends its data to a neighbour node, where they are combined with other sensor readings or just retransmitted to other nodes until it reaches the sink. The sink usually has more capabilities than normal nodes and can communicate to the users through Internet connection or other means.

There are various applications of multi-hop WSNs which include, but are not limited to the following: agriculture, infrastructure protection (e.g., water distribution and power grids), health care, environmental monitoring (e.g., habitat, traffic, security, etc.), fire protection, warehouse management, smart transportation, water and waste management, military applications, biomedical sensor engineering, and so on. In such applications a single static sink is normally used. Some applications may use more than one sink or they may even use a mobile sink. Whether mobile sink or static sink, single or multiple sinks, the role of a sink is to collect sensed data from sensors and sometimes analyse and process the data for specific applications. For ease of access the sink might be connected to the Internet so that the collected data can easily be accessed by the users at will.

3.7 Periodic, On-Demand and Event-Driven Reporting

There are three data communication modes followed in WSNs namely: Periodic, on-demand and event-driven reporting. Each communication mode has its own application in WSNs. For periodic reporting (or time driven) mode, information is collected by sensor nodes from the environment at scheduled times and periodically transmitted to the sink. For the event-driven mode the data is collected by the sensors once an event (e.g. movement) has been detected and is then transmitted to the sink. For on-demand (or query-driven) mode, as the name indicates, the user decides when and what data to collect and from which part of the network.

Data can be gathered from an event occurring in a particular part of the network or it can be gathered periodically from the whole network. Event reporting is more suitable for applications whereby data is generated by events in the network while periodic reporting is suitable for a periodic reporting of a chosen phenomenon throughout the target WSN. Applications using periodic reporting are less sensitive to time than applications using event driven reporting.

Once the data has been collected from the environment then it is transmitted using the underlying routing protocol in the WSN. However, the data can be fused with other collected data from other sensors as it traverses the network or on the other hand it can be sent without any modification on the actual data. This leads to two concepts namely data gathering and data aggregation. With data gathering the data is not modified as it traverses the network. The issue with such mechanism is that it can bring redundant data and hence energy wasting. This inefficiency is addressed by data aggregation which combines data from different sensor nodes before sending it. This does not only eliminate redundancy but it also reduces the number of transmissions hence saving the network of its valuable and limited energy resource.

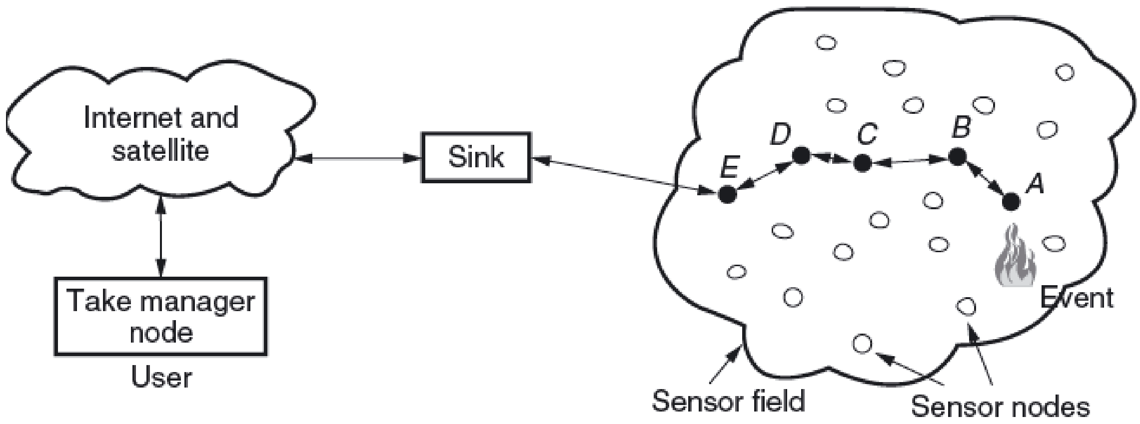


Figure 3. 9 A scenario of event-driven reporting (Nayak and Stojmenovic 2010).

For example in an event-driven application as shown in Figure 3.9 the server transmits the data to the sink by using either data gathering or data aggregation and a routing protocol. In this example the event is a fire which is then detected by the node A. Once node A detects the fire it sends the data to the sink as shown (Nayak and Stojmenovic 2010).

3.8 The Protocol Stack

A protocol stack is an architectural model that provides a common frame of reference for discussing WSN communications. This model is also used as a guide to develop a sensor node and its protocols. Each of these layers performs a specific function in the process of communicating over the WSN. A layer defines a data communication function which may be performed by any number of protocols.

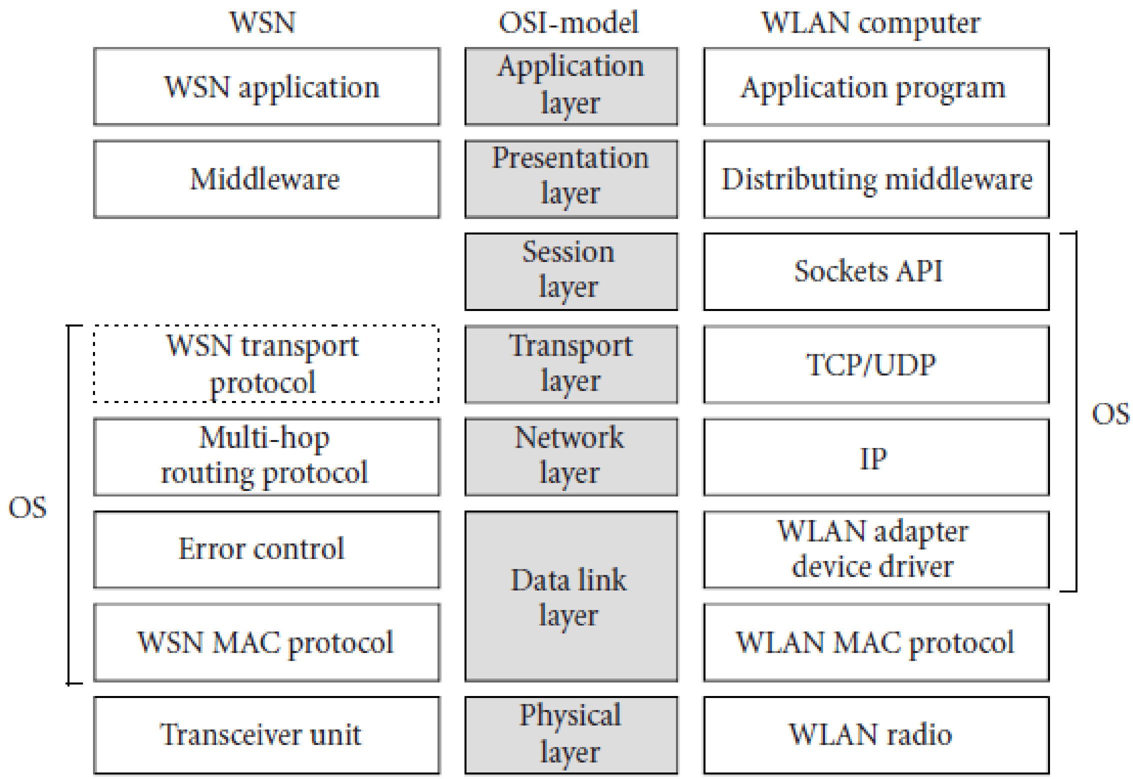


Figure 3. 10 OSI model, WSN, and distributed system in WLAN protocol layers (Kuorilehto et al. 2005).

A protocol X in layer Y in node A will communicate to its peer protocol in node B in layer Y. A peer is an implementation of the same protocol in the equivalent layer on another remote node. There is a standard for successful peer to peer communication.

In theory, each protocol is only concerned with communicating to its peer i.e. it does not care about layers below or above it. However, in practice there is a dependency between layers in the same node. In the process of sending data from a local application to an equivalent remote application, the layers must agree on how to pass data between themselves in a node. Therefore upper layers rely on the lower layers to successfully transfer the data across the underlying network.

Here is the list of the responsibilities of the different layers and the protocols that might operate there.

- **Physical Layer**

  Responsibilities

  ✓ Frequency selection

- ✓ Carrier frequency generation
- ✓ Modulation
- ✓ Signal detection
- ✓ Encryption

- **Data Link Layer**

Responsibilities

- ✓ Medium access
- ✓ Data frame detection
- ✓ Multiplexing of data streams
- ✓ Error control

Protocols

- ✓ Mac for sensor networks
- ✓ CSMA-Based Medium Access
- ✓ Hybrid TDMA/FDMA-based
- ✓ Self-organising Medium Access Control for Sensor Networks (SMACS) and the Eavesdrop-And-Register (EAR) Algorithm

- **Network Layer**

Responsibilities

- ✓ Routing
- ✓ Datagram network service

Protocols

- ✓ Flooding
- ✓ Gossiping
- ✓ BRALB
- ✓ C-BRALB
- ✓ Low-Energy Adaptive Clustering Hierarchy (LEACH)

- **Transport Layer**

Responsibilities

- ✓ End-to-end Reliability
- ✓ Multi-hop retransmission
- ✓ Congestion
- ✓ End-to-end Security
- ✓ Like SSL: authentication, encryption, data integrity

Protocols

- ✓ TCP
- ✓ UDP

- **Application Layer**

Responsibilities

- ✓ Data base queries
- ✓ Sensor network management
- ✓ Time synchronisation/calibration.

Protocols

- ✓ TELNET
- ✓ HTTP
- ✓ FTP

3.9 WSN Applications

The wide range of affordable sensors and collective instrumental functionality of WSNs, coupled with the underlying wireless networks, create a possibility of unprecedented levels of data access and associated intelligence, creating a new dimension of application for various industry sectors. However, all these applications can be categorised into three major groups namely: space monitoring, which include the objects as a part of the space; state of object operation monitoring; and the interactions between object and space. The first group space monitoring is the most common and basic use of WSNs (dealing with the physical environment) whereas state of object operation monitoring group deals with specific entity rather than the surrounding of the entity. The third group; interactions between objects and space group deals with the complex monitoring and control over communications, and interactions between objects, and between an object and its surrounding. It is true that WSNs were conceived with military applications in mind, which includes tracking of enemy activities and battlefield surveillance. However, there are various civil applications as well which undoubtedly outnumber the military ones and are applicable to many situations.

With both the military and civil interests in WSNs it is no surprise that there are large numbers of applications that exist for WSNs. Some of these applications include security and tactical surveillance, weather monitoring, distributed computing, detecting ambient conditions such as temperature, movement, sound and light, vibration, gases, smoke or the presence of certain chemical or biological objects. Some of these applications and their underlying WSN structures will be discussed in the next section.

3.9.1 Environmental Monitoring

With large number of sensor nodes deployed in a chosen geographical location, large amount of interest datasets can be collected over time and some possible patterns and trends can be obtained from the dataset collected. In such situations the phenomenon to be sensed includes temperature, light, microclimate monitoring, traffic monitoring, pollution level monitoring and indoor climate control.

In the USA over 6000 landfills are being used to dispose household solid waste and non-hazardous industrial waste such as sewer sludge, construction debris and household waste within these landfills may cause the waste to undergo biological and chemical reaction such as fermentation, oxidation-reduction and biodegradation. This process may cause harmful gases like nitrogen, carbon dioxide, methane, sulphide compounds and ammonia to be produced. These harmful substances may escape into the environment by leaking causing pollution and diseases. These substances may also contaminate the ground water up to 100 of metres from the landfill, causing serious problems (Cordeiro and Agrawal 2006).

With these serious implications there is a dire need to monitor landfills. Unfortunately, the current method of landfill monitoring is not very scalable as it employs periodic drilling of wells to collect gas samples and analysing them off-site which requires a lot of time. WSN can provide a solution to such a dilemma by interfacing gas sensors with custom-made devices and wireless radio, and transmit the sensed data to the sink for further analysis (De Morais Cordeiro and Agrawal 2006). Large number of sensor can be deployed within a targeted area thereby allowing the real time monitoring of pf gases being emitted by the landfills. With appropriate type of sensors various hazardous pollutants such as ozone, sulphur dioxide or carbon monoxide, and lead can be monitored (Agrawal et al. 2004).

3.9.2 Habitat Monitoring

There are various applications for WSNs in habitat monitoring. WSNs are used to monitor a phenomenon of a habitat such as temperature, humidity and vibration. Sensors can also be used to count wildlife in a given area.

In Canada in the WMU Area 47 Ontario North of Parry Sound district a survey of the White Tail Deer (Odocoileus Virginianus) (Tovar et al. 2010) is done using a technique called “Packet Group Count”. The data acquired help Ontario’s citizens to open or close the hunting season. This is done manually in two parts. In the first part a helicopter is used to locate a $1km^2$ topographical map to stratify the area. After this some plots are selected based on the expected density of the deer per stratum. Each sampled plot is an equilateral triangle of 1km per side. In the second part each triangle side is divided into five pellet group subplots, each 40m long and 2m wide. Two person ground crew surveys the strata by walking all the three sides of the triangle plot looking for a dead deer. Pellet group counts are evenly spaced intervals. All the subplots are searched and all deer pellet groups seen are counted. After the survey the number of pellet groups counted (and the total area searched for pellet) is added up for each stratum. The data collected from this survey is used to obtained an estimation of the deer population for the total area (Konze 1998).

In order to solve the limitations of such a survey, Tovar et al. (Tovar et al. 2010) proposed to use a wireless DTN (Delay Tolerant Network) sensor network. The DTN sensor network is used to acquire the required data. A data mule (automobile or helicopter) is then used to harvest the data from the principal mode via DTN intermittent schedule contact from a node to the site. A WSN network is then put in place in each of the key located areas. A key area is a place that has already been marked as having higher number of White Tailed deer. It is the duty of the mule to trace a route to harvest data of each principle node. With this system, data can be regularly harvested with less time and less human intervention.

In August 2002 some researchers from the University of California at Berkeley (UCB) and Intel Research Laboratory deployed a mote-based layered sensor network in Great Duke Island (GDI), mainly to monitor the habitat behaviour of storm petrel (Mainwaring et al. 2002). A gateway and 32 motes are strategically located in the targeted area. The targeted area is grouped into sensor patches and the data collected from there is transmitted to a gateway (cerf cube). The gateway then forwards the information from the sensor patches to a base station which then provides data logging and replicates data periodically every 15 minutes over a satellite link to a database in Berkeley via a satellite link. The data is accessed locally by using a small PDA-sized device to perform local interactions or remotely via the replica database server in Berkeley (Cordeiro and Agrawal 2006).

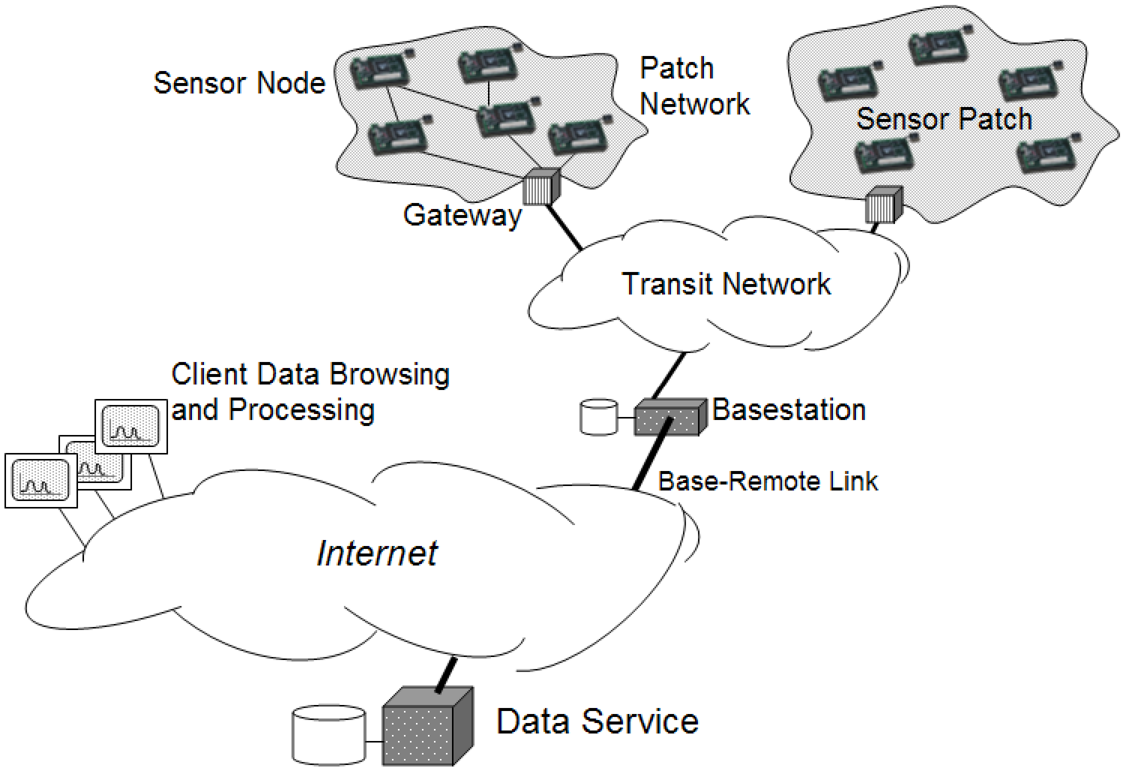


Figure 3. 11 System architecture for habitat monitoring (Mainwaring et al. 2002).

3.9.3 Body Area Network

Sensors have a wide range of applications including human healthcare applications. Specialised sensors and transducers are developed and deployed to measure human body conditions and features such as body temperature and heartbeat so as to help predict human conditions accurately and efficiently. Wireless body area networks (WBANs) has its roots from personal area network (PAN), a term originated from Zimmerman (Zimmerman 1996) and was then developed by IEEE P802.15 working group (Heile et al. 1999). Philips were among the first to use the term BAN instead of PAN. During that time BAN was normally considered under PAN. Philips used BAN with added distinct features (Jovanov et al. 2009). As BAN evolved from connectivity of personal electronic consumer good for convenience to the user to healthcare and medical application, there has been an increased interest in BAN and various proposals have recently been introduced (E Jovanov et al. 2009;Roy 2003;Schmidt, 2002). A design for wearable sensor vest for measuring and transmitting physical features such as heart rate, movement and temperature is proposed (Kimel 2005).

Chen et al. (2004) propose a method of measuring physiological, environmental and behavioural parameters by using audio sensors accelerometers and electrical signal in order to figure out the daily human life pattern. These instruments help to gather the data from the sensors then refining and segmenting it for final processing and interpreting for the event detection.

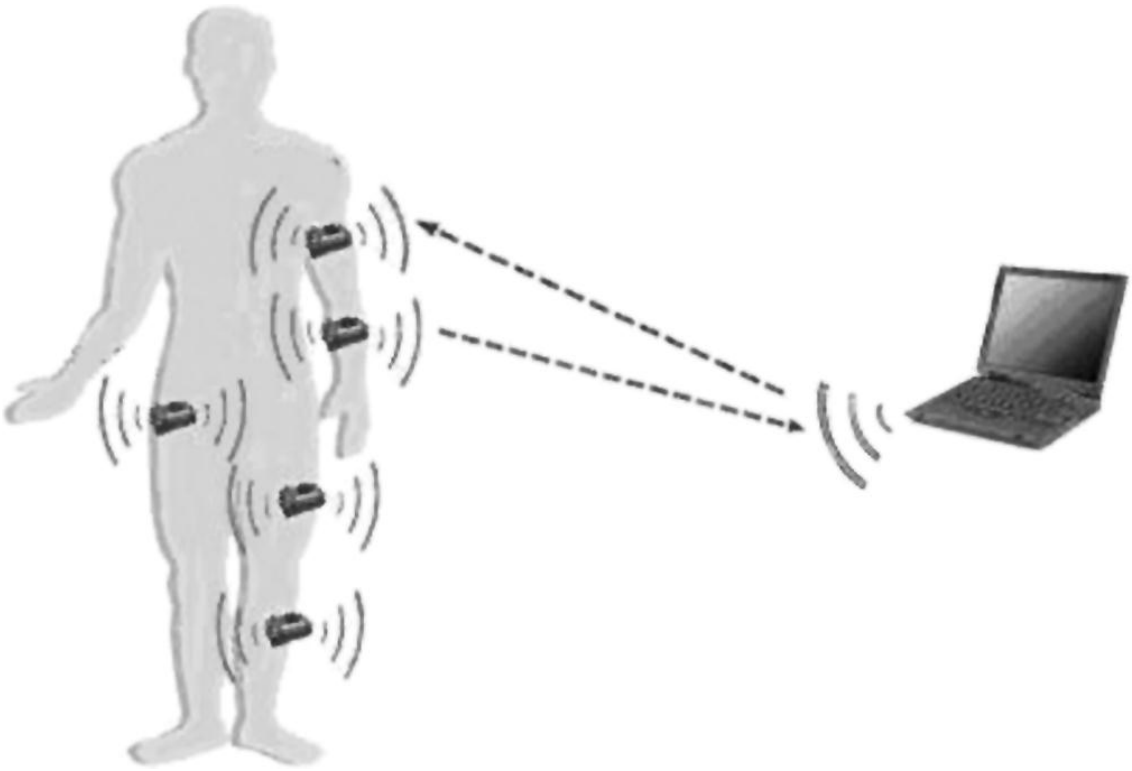


Figure 3. 12 Monitoring limb movement in stroke patient rehabilitation (Nayak and Stojmenovic 2010).

3.10 Modelling Sensor Networks

In general system modelling is an act of representing an actual system in a very simple way. Mathematics or Statistics can be used to abstract the key parameter of the system and represent it by equations. System modelling gives an idea of how a system would work and behave if actually implemented which is very vital in system design and development. With modelling various parameters of the system can be changed, tested and analysed so as to get an optimum performance thereby saving both time and cost in system development. To model a system, the important characteristics or properties of the system must be put into consideration while making simplified assumptions on less important properties based on the objective of the modelling. Assumptions can be made to simplify the modelling but unrealistic or too many assumptions will surely lead to an inaccurate representation of the overall system. Any algorithm for WSN connectivity that does not put interference into consideration may work very fine in theory but may be inefficient or even totally wrong in real world. It is therefore, important to be very careful when making assumptions.

In general there are two traditional modelling approaches: analytical approach and simulation approach. In the analytical approach the system to be modelled is studied and then mathematical tools such as queuing and probability theories are used to represent and describe the system. In general such an approach is more applicable to simple and relatively small systems. It is very cost effective and the numerical solutions to this type of methods do not require heavy computational efforts. Again the assumptions that should be made using this model needs to be realistic or else the results might not be accurate. If realistic assumptions are made, then it can provide an abstract view of the working mechanism of the overall system and also the interaction of individual components of the system.

Simulation approach is used to model systems which are very large and complex such that the mathematical abstraction of the model is almost impossible. It is used to model applications in business analysis, manufacturing, engineering research, biological experimentation and many other areas. With simulation it may be possible to put almost all the details of the system into the simulation models so as to get better results. It is necessary to put only the relevant information as part of the simulation model as too much information could end up getting huge simulation time and high computational effort (Issariyakul and Hossain 2009 ; Boukerche 2008). In this thesis both approaches are used so as to get optimum results. The proposed algorithm was first analysed mathematically and proven to be energy efficient and reliable. This was further followed by a simulation of the algorithm which also conforms to both the mathematical analysis and predictions.

A simulation is a structure of components namely: entities, activities and events, resources, global variables, a random number generator, a calendar, system state variables and statistics collectors (Issariyakul and Hossain 2009).

Entities have attributes which are characters that make them unique. Entities are objects with unique characters and they interact with one another in a simulation program to bring changes to the system program. In wireless sensor networks entities can be sensor nodes, packets, flows of packets or even simulation clock. If a data packet is an entity then its attributes may be sequence number, packet length, priority and the header.

Resources are part of a complex system; they are limited and are to be shared by some of the entities depending on the simulation. In WSNs resources could be bandwidth, communication channel etc.

When entities engaged with one another or do something it is called an activity and thus creates an event which may trigger some changes of the system states. The major types of activities in WSN simulators are delay, queues and logic.

For WSNs a global variable example could be the energy of the nodes. A global variable can be called or accessed by any function or entity in the system, which keeps track of some common or shared parameters of the simulation.

A Random Number Generator is used to generate random numbers and can be used to create randomness in the simulation model. For example, if a certain process possesses a uniformly distributed delay say between 10 mins and 20 mins, as an entity goes through the process, the random number generator would generate a number between 0 and 1. This random number is then fed into the uniform distribution that has a minimum of 10 and a maximum of 20. If the function is $f(x) = 10 + (x)(20-10)$ where x is the generated random number say 0.7312 then f(x) = 10 + (0.7312)(20-10) = 17.312 delay (Ingalls 2008).

System state variables are variables built within a simulation package. A common system state variable is the current time of the simulation which is updated every time an entity is taken from the calendar. The statistics collector collects data generated from the simulation such as global variables (e.g. for WSN), or certain performance statistics based on attributes of the entity so that meaningful inferences can be drawn from such data (Ingalls 2008).

3.11 WSN Security

Security is very vital for any network and WSNs are no exceptions. The progress registered in security in WSNs is not as fast as the progress registered in routing algorithms for WSNs. This could be among other factors attributed to the fact that a security solution in most networks requires some high computational power, which is a scarce resource in WSNs. Security for WSN is still an open research issue and it is most important for WSN in military surveillance applications.

3.12 Summary

In this chapter, a typical sensor node and the features affecting the energy resources of a WSN were discussed. The three types of data communication: periodic, on-demand and event-driven were also discussed together with single-hop and multi-hop data transmission method. Various application types of WSNs were also reviewed. The information provided in this chapter forms an important background to the energy efficient routing algorithms proposed and evaluated in this research work.

Chapter 4

ROUTING PROTOCOLS

Chapter 4

ROUTING PROTOCOLS

4.1 Introduction

Wireless sensor network is a network of large number of sensor nodes that are deployed in a target area to monitor different types of physical or environmental conditions such as vibration, temperature, motion, sound or even pollutants (carbon mono oxide) at various locations. Usually the important data that has been sensed by the sensors need to be collected and analysed by users (Sinha and Barman 2012). For this to happen there is the need for a set of instructions of how to collect the data and transmit it to the sink. The set of instructions of how to transmit data is simply a routing protocol. Routing in WSNs is not the same as in MANETS whereby each node always has a globally unique ID which is either its MAC address or IP address, thereby making the routing problem address centric. In WSNs, nodes are very small with huge resource limitations and mostly deployed in the order of thousands hence unique IDs are not always available for individual nodes. This can create problems in distinguishing nodes.

A WSN with no node IDs can create problems in certain applications whereby there is the need for a label on the collected information for effective processing. One good scenario is the Glacier Bed Deformation monitoring System (Ye et al. 2005) where the temperature is sensed and reported by sensors periodically. The collected temperature readings will make no sense without the knowledge of the data source. Therefore, an ID scheme for WSN is not only important from an application standpoint but also for the effective operation of the system as most routing protocols rely on source and destination ID with the exception of flooding.

A node ID scheme is assumed in most of the research publications in routing algorithm for WSN (Wu and Tseng 2007). A node ID scheme for a sensor can be classified into two namely: Global or Local. The global unique ID is mostly given during manufacturing process and is burned into the system similar to 48-bits MAC address in routers, switches or computers. With a large number of nodes being deployed in a WSN, a global unique ID needs to be very long thereby producing large overhead in the resource limited tiny sensor nodes. WSNs are mostly deployed in harsh environment with the tendency of high interference into the radio communication channel. In order to avoid or minimize transmission errors most WSNs support short packets. The Berkeley motes have a fixed packet size of 36 bytes (Dam and Langendoen 2003). If a packet is being transmitted with a packet header containing only the source and the destination node IDs without any additional control information using a Zigbee standard ID of 64 bits (ZigBee), the overhead for the source ID is 64bits/8 = 8 bytes likewise the destination ID, making a total of 16 bytes overhead. With this (16/36)*100 approximately 44% of the total size of the packet is just used for the ID overhead leaving only 65% for the data. Hence in order to decrease the ID overhead a *16-bit short address field is used in Zigbee* for packet transmission (Wu and Tseng 2007). With this it is clear that global ID for most WSNs is not feasible. A more feasible scheme is to use a local ID scheme.

In the local ID scheme each node is assigned an ID before or after the deployment. The generation of location ID can be automatic and without any centralised control mechanism. This local ID is generated with the minimum number of bits necessary. The generation of local ID is discussed in (Akyildiz 2002; Culler et al. 2004) No matter what method of node ID generation is used, the ID algorithm should be efficient in terms of overhead, time, energy and storage and also be fully distributed. The ID establishment algorithm can also be classified as location-independent or location dependent. The location-independent ID algorithms are flat and are sometimes generated randomly while the location-dependent algorithms are generated using location information.

4.2 Routing Scenarios in WSNs

There are various patterns of communication within a WSN. They can be categorised into two namely: sink-to-sensors and sensors-to-sink. These patterns of communication determine the routing protocols to be used in the network. Hence the understanding of these communication patterns is crucial to the design of any routing algorithm. These patterns include sink-to-sensor communications and sink-to-all communication.

4.2.1 Sink-to-Sensor Communications

With this type of communication it is the sink that always initiated the communication to a sensor or a group of sensors. This category of communication is further divided into sub-categories, which will be discussed below. A sink to one is a subset of this category, and as the name implies it is a scenario whereby a sink targets a particular node in the network. There are various schemes for achieving this, such as broadcasting the message with a destination ID of the targeted sensor node. This is not an efficient way of targeting an individual sensor; an efficient solution is to transmit the message directly or send it through its neighbouring nodes. With sink-to-one communication a globally unique ID makes the communication easy, otherwise locating the destination node becomes a real issue (Wu and Tseng 2007).

A sink-to-all communication is a type of sink-to-sensor communication. In this communication pattern two scenarios are used. In the first scenario it makes use of flooding where the messages are duplicated from one node to another which leads to all the nodes getting the "same" copy of the messages. Queries use this form of communication most of the time. The other communication scenario makes use of in-network packet processing of results thereby avoiding duplicate transmission of data. Some routing protocols use this when building the shortest path length (e.g. hop counts) from a sensor to the sink (Ye et al. 2001). In this scenario a node processed received data before it is sent out again.

A sink-to-region communication is used in applications where the sink is required to communicate to one or a set of sensor nodes in respect of their locations. An example could be a sink wanting to get the temperature reading of sensors in a particular room or an identified area. One way of achieving this is to send the packet to one of the nodes within the interested region; the recipient node can then flood query to all the nodes within the region.

A sink-to-subset is different from the previous communication pattern in that the targeted sensors are not grouped into one area but scattered all over the network. In such scenario the sink has no clue where the nodes possessing the required data are in the network, so the only solution is blind flooding the queries.

The last communication pattern under this category is the All-to-Sink communication which is used in applications where all the sensors are required to report the collected information periodically.

4.2.2 Sensor-to-Sensor Communication

In this scenario of communication, data is moved from the originator nodes to intermediate nodes for in-network processing so as to save communication overhead. With the in-network processing, the data becomes more meaningful than the raw sensor reading. A WSN that uses either in-network data processing or data-centric storage systems needs sensor-to-sensor communications. *However, this communication pattern uses more overhead on local storage, on energy, and control packets than other communication pattern type*s.

4.3 Routing Protocols in WSN

In general, routing protocols can be divided into two categories: routing based on network structure and on protocol operation. The network structure based routing is further sub-divided into: Flat-based routing, Hierarchical-based routing and Adaptive-based routing. The protocol operation category is further sub-divided into: Negotiation-based routing, Multipath-based routing, Query-based routing, and Location-based routing. However, it is worth mentioning that there are protocols that are hybrids of different groups mentioned above.

4.3.1 Network Structure Based

This category of routing protocols is classified based on the structure of the network. It is further divided into three sub-categories.

4.3.1.1 Flat-based Routing

In this sub-category of routing protocols all nodes are assigned equal role. Some of the routing protocols that fall under this category will be discussed here.

In Directed Diffusion (DD) routing protocol (Intanagonwiwat et al. 2000) schemes like data aggregation and dissemination are employed so as to eliminate redundancy thereby reducing the number of transmissions; thus saving the network energy and maximising its lifetime. DD routing protocol makes use of multiple paths to the sink and allows in-network consolidation of redundant data. In DD the sink diffuses Interest messages to all sensors by way of flooding. The Interest messages describe a task to be done by the network and are flooded throughout the network in hop-by-hop fashion. A node that receives an Interest will broadcast it to all its neighbours, in this way the Interest is flooded throughout the entire network. With the propagation of Interest messages, *gradients are made to draw data satisfying the query towards the requesting node*. The sensor nodes setup gradients toward the nodes they received Interest from. This process completes when all gradients are setup from the data sources back to the sink. A source node may have different paths with different gradients towards the sink, however, the best paths are selected so as to reduce further flooding. Figure 4.1 depicts an example of the operation of directed diffusion. However, it is clear that DD generates a large number of redundant data; therefore DD has a poor performance.

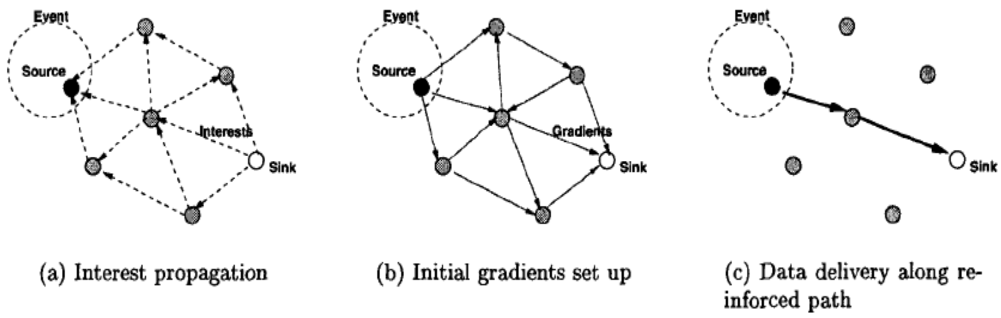


Figure 4. 1 A simplified schematic for directed diffusion (Intanagonwiwat et al. 2000).

Improved directed diffusion (IDD) was proposed to solve the poor performance issue. Improved directed diffusion consists of five phases namely cluster formation protocol, Interest diffusion, data propagation, reinforcement and the final routing maintenance and update phase. IDD groups the nodes into different clusters, with few nodes as cluster heads and the remaining nodes as cluster members. Interest is flooded only among the cluster heads. Cluster members sense and transmit the sensed data to their respective cluster heads. A cluster head compresses and aggregates the data received from its cluster members and sends this sensing data to base station along a reinforcement path. Simulation results show that IDD is more energy efficient than DD. It also reduces the data transmission delay and prolongs the network life time (Yanrong 2007).

4.3.1.2 Hierarchical Routing

The main purpose of this sub-category of network structure based routing protocol is to achieve a scalable and efficient communication. In hierarchical schemes, the nodes are grouped into clusters by a clustering scheme. In each cluster there is a cluster head CH which is usually a node with higher energy and will have a higher processing power as well. The CH is responsible for aggregating and processing data from nodes within its cluster and transmitting the data to the sink. The low energy nodes which reside in a cluster are used to sense the network for the targeted phenomenon to be monitored and then transmit the sensed data to their cluster head. In the cluster based schemes, the creation of clusters and the appropriate role assignment of selected tasks to CHs are geared to improve overall system lifetime, scalability and energy efficiency. A good example of such category of routing protocols is the DDBCI discussed previously.

4.3.1.3Adaptive Routing

In Luwei et al. (2011) SPIN-1 is proposed which is a family member of adaptive protocols called Sensor Protocol for Information via Negotiation (SPIN).SPIN-1 uses ADV messages to advertise a new data. SPIN-1 is a modified version of SPIN, with its sole aim to address the problems of SPIN such as the “blind forward” problem and the “data inaccessible” issue. The “blind forward” is caused by the fact that when a node has some data to send, it broadcasts ADV message to its neighbours and will send the data to all its neighbours that respond. The neighbours will also repeat this process until the data reaches the sink. It is the same forwarding process in SPIN that leads to the “data inaccessible” issue. As mentioned earlier, with the flooding of data packet the nodes closer to the sink are bound to take more tasks in forwarding the data to the sink and may be depleted off their scarce energy resource leading to partitioning of the network.

Similar to SPIN, SPIN-1 protocol is based on the following assumptions of the network model:

(1) *The initial energy of each node is the same*. A node and its neighbours can communicate with each other and the *communication link is symmetrical*,

(2) There *is no interference* to the communication between two nodes, and the nodes are immobile.

(3) There is coverage and no partitioning of the network at initial stage.

(4) The wireless signal uses same energy in all directions.

The working mechanism of SPIN-1 is based on the above assumptions and it is a process that involves a three-phase handshake, namely data broadcasting stage, data requesting stage and data transmission phase.

In the data broadcasting stage, a node with a new data to send or forward will broadcast an *ADV message* to its neighbours, and set a timer. The properties describing the data are contained in the ADV metadata.

The next stage to the broadcasting stage is the data requesting stage. There is an energy threshold value for a node to take part in the three-way handshake. A node that received an ADV message will first check its energy value; if it is below the threshold value then it will not respond. If above the threshold, then it checks if it has already received the data, if it has, then the REQ flag bit is set to 1 and its energy is updated in the REQ as well and sent to the source node. In SPIN protocol the source is not notified when the neighbour has data while in SPIN-I the source will be informed accordingly. On the other hand when a node received an ADV message and passes the threshold test and does not have the data before, it will set the REQ flag to 0, and update its energy value using the REQ message and send the REQ message to the source.

After the data requesting stage the next stage is the data transmission phase. The REQ message information is used by the source node to update its neighbour list record. The source nodes use the threshold time and the flag bit status in order to transmit the message to a neighbour. The flag bit it received might be 1 or 0 or a combination of both. In the case that it is a combination of 1 or 0, then it will select the neighbour with a flag bit set to 0 and has the highest energy in case of a tie, it will select one at random. It does the same thing when the flag bits are all 0. In the case that the received flag bits are 1 and the time is longer than the threshold it results in the "data inaccessible" problem, the source nodes will then select the node with the highest energy value and mandatorily forward the data to it. To update its neighbour list, it then trims off all the nodes that do not respond to its REQ message from its neighbour list.

In order to mathematically analyse and compare SPIN-1 and SPIN, it is assumed that a m-bytes of message is being received by a node and then retransmitted to a neighbour node. It is further assumed that both ADV and REQ messages are L bytes and that the energy cost to send or receive a byte of data are E_m and E_r units of energy respectively. The average number of neighbours for a node is N. It is also assumed that the network is distributed *with no packet loss or queuing delay*. A node forwards the m byte of data it receives to the next hop node.

(1) The steps involved when node B forwards the M byte of data using the SPIN protocol are:

(1a) Energy consumption of sending ADV message to a node is $(N\text{-}1)LE_m = E_m(LN\text{-}L)$

(1b) Energy consumption of receiving REQ message from N-1 neighbour nodes is $L(N\text{-}1)E_r = E_r(LN\text{-}L)$

(1c) Energy consumption of sending m bytes of data plus L bytes (meta data contains energy value and data type) is $(m+L)(N\text{-}1)E_m = E_m(mN+NL\text{-}m\text{-}L)$

Therefore the total energy of node B forwards the m bytes of data is:

$$1a + 1b + 1c = E_m(LN - L) + E_r(LN - L) + E_m(mN + NL - m - L) =$$

$$E_r(LN - L) + E_m(2LN + mN - m - 2L) \tag{4.1}$$

(2) The steps involved when node B receives m bytes of data using the SPIN protocol are:

(2a) Energy consumption of receiving ADV message is LE_r

(2b) Energy consumption of sending REQ message is LE_M

(2c) Energy consumption of receiving m bytes of data plus L bytes (meta data) is $(M+L)E_r$ $= E_r(m+L)$

Therefore the total energy of node B receives the M bytes of data is:

$$2a + 2b + 2c = LE_r + LE_m + E_r(m + L)$$

$$LE_r + LE_m + E_r(m + L) = 2LE_r + mE_r + LE_m = E_r(2L + m) + LE_m$$

$$E_r(2L + m) + LE_m \tag{4.2}$$

From the above calculations the minimum energy assumption for node B receiving m bytes of data and forwarding the data to its neighbours in SPIN is obtained from Equation (4.1) + Equation (4.2).

$$E_r(LN - L) + E_m(2NL + mN - m - 2L) + E_r(2L + m) + LE_m$$

$$E_m(2LN + MN - m - 2L) + LE_m + E_r(LN - L) + E_r(2L + m)$$

$$E_m(2LN + MN - m - 2L + L) + E_r(LN - L + 2L + m)$$

$$E_{SPIN} = E_m(2LN + MN - m - L) + E_r(LN + m + L) \tag{4.3}$$

With the above assumptions the energy consumption in SPIN-I when transmitting the same m bytes of data is shown here. It is also assumed that the bit length of energy value and flag carried by the REQ is very small and is negligible.

(3) The steps involved when node B forwards m bytes of data to its neighbour using the SPIN-I are:

(3a) Energy consumption of sending ADV messages is $(N\text{-}1)LE_m = (N\text{-}1)LE_{m;}$

(3b) Energy consumption of receiving REQ message from N-1 neighbour is $L(N\text{-}1)E_r$

(3c) Energy consumption of sending m bytes of data plus L bytes (metadata) is (m+L)

Therefore the total energy of node B forwarding m bytes data to its neighbours is

$$3a + 3b + 3c = E_m(NL - L) + E_m(M + L) + E_r(LN - L) \tag{4.4}$$

The procedures and energy cost of node B receiving m bytes of data in SPIN-1 and SPIN protocol is the same. Therefore the energy cost of node B receiving m bytes of data in SPIN-1 is equals to equation (4.2) = $E_r(2L + m) + LE_m$.

From the calculations, the minimum energy of node B receiving and forwarding m bytes of data to its neighbours in SPIN-1 is obtained from equation (4.2) + equation (4.4) :

$$E_{SPIN-I} = E_m(NL - L + m + L) + E_r(LN - L) + E_r(LN - L) + E_r(2L + m) + LE_m$$

$$= E_m(NL - L + m + L + L) + E_r(LN - L)$$

$$+ E_r(LN - L) + E_r(2L + m) \qquad = E_m(NL + m + L)$$

$$+ E_r(L + NL + m)$$

$$E_{SPIN-I} = E_m(NL + L + m) + (L + NL + m)E_r \tag{4.5}$$

The energy difference between E_{SPIN} and E_{SPIN-I} is equals to E_{SPIN} - E_{SPIN-1} =

$$E_{SPIN} - E_{SPIN-1} = E_m(2LN + MN - m - L) + E_r(LN + m + L) - (E_m(NL + L + m)$$

$$+ (L + NL + m)E_r)$$

$$E_{SPIN} - E_{SPIN-1} = E_m(2LN + MN - m - L) + \cancel{E_r(LN + m + L)} - E_m(NL + L + m)$$

$$- \cancel{(L + NL + m)E_r}$$

$$E_{SPIN} - E_{SPIN-1} = E_m(2LN + MN - m - L) - E_m(NL + L + m)$$

$$E_{SPIN} - E\_(SPIN - 1) = E_m(LN + MN - 2m - 2L)$$

$$E_{SPIN} - E_{SPIN-1} = E_m(N(L + M) - 2(m + L))$$

$$E_{SPIN} - E_{SPIN-1} = E_m(L + M)(N - 2) \tag{4.6}$$

In equation 4.6 E_m is always a positive value and the same is true for (m+L). From this equation, it is clear that only when each node has one neighbour is the energy consumed by SPIN-1 greater than that for SPIN, although this scenario is impractical. For N equals 2 the two protocols use the same energy and even such a scenario is very unlikely as the nodes need to be connected in a chain-like form. For any value greater than 2, equation 4.6 becomes positive indicating that the energy consumption of SPIN-1 is smaller than that of SPIN. The simulation results also confirm these findings.

4.3.2 Protocol Operation Based Routing

This category of routing protocols is classified based on the operation of the routing protocol itself. It is further divided into four sub-categories which will be discussed below.

4.3.2.1Negotiation-Based Routing

In negotiation-based routing, as the name implies, the communication between nodes is negotiated based on the available resources. It uses high level data descriptors so as to avoid unnecessary transmission of redundant data thereby increasing the overall network lifetime. SPIN and SPIN-1 are two classic examples of such protocols, which have been discussed in the previous section. Both SPIN and SPIN-I employ a scheme of detecting and avoiding the transmission of redundant data to the sink.

4.3.2.2 Multipath-Based Routing

In WSN the number of nodes are most of the time numerous. As a result there is always the likelihood of multiple paths from the source to the sink. This is very beneficial in a network as it helps to balance the traffic among the multiple paths and as a result creates fault tolerance within the network. It is for this reason *most routing protocols in WSNs keep and maintain multiple paths to the sink*. The maintenance of multipath in WSNs comes with a cost of additional overheads; however, the benefits outweigh the short comings. In WSNs that employ multipath, there is always a fast switchover to the secondary path when the primary path fails thereby increasing the reliability of the network. There is no delay associated with moving from the primary path to the secondary path thus making such protocols ideal for applications sensitive to delay.

Multipath routes can either be defined as node-disjoint or not node-disjoint. A set of multiple paths between a source and a destination node are said to be node-disjoint when there is no node overlap amongst their various paths. However, in reality most paths in such routing protocols are not node-disjoint as they most likely have one or more common nodes along the paths. Therefore an alternate path in this discussion can either be node-disjoint or not.

Multipath routing was used in Wang and Qi (2008) to enhance the reliability of the WSN; load balance the traffic among the nodes and hence increase the WSN lifetime. In this algorithm the nodes are assumed to be static, have the same energy and capabilities except the *sink* which has unlimited energy and *is located at the centre of the WSN*. In this algorithm node connectivity is based on the distance between them and if it is below a certain defined threshold value the nodes are said to be connected. This algorithm employs Dijkstra by finding the shortest path between the source and the destination. Once the Dijkstra algorithm finds the shortest path from the source to the destination, then the nodes in that path are marked so as to avoid using the same nodes in finding an alternative path between the source and the sink. *After obtaining the first path, a second alternative path is searched by using the same rule. When the second path is obtained the nodes are marked and then the final path is searched by using the same method*. Once three paths are identified the data is load balanced along the three paths from source to sink. This algorithm is very similar to OSPF (Open Shortest Path First algorithm) used in the Internet. OSPF can load balance on equal path cost by default up to 4 paths which can be configured up to a maximum of 16 paths. The multipath algorithm was simulated and compared when only one path is used. The algorithm was more efficient when used in multipath mode.

In (He 2010) a multipath routing algorithm is proposed with the following assumptions in the system model. The WSN is composed of scattered sensor nodes and a base station. Each node has a unique identifier (ID) and is able to directly communicate with its neighbour in a bidirectional manner. The sensors have a limited energy and are aware of their energy value.

The essence of this protocol is similar to the previously discussed multipath protocol in that they both discover multiple paths from the source to the sink node so as to load balance the traffic along these paths. The major difference between these protocols is the path discovery process. In this protocol paths are not discovered using the Dijkstra algorithm but by sending a random message by the sink which randomly traverse the network. At the same time the source node also does the same thing. The two messages which act as discovery path will meet after sometime. From the footprint of the two discovery messages the path from the source to the sink is forwarded.

Once the first path has been established the source node starts to discover an alternate path by sending a message to one of its neighbours which is not in the first path. This message *is randomly transmitted* in the network till it reaches one of the nodes discovered in the first path. Thus, another path is constructed from the footprint of the second message together with that of the other node. *This process is repeated to create the needed paths*. The other difference from the previous algorithm is that in the previous algorithm node-disjoint paths were created whereas in this one there is node overlapping on alternate paths.

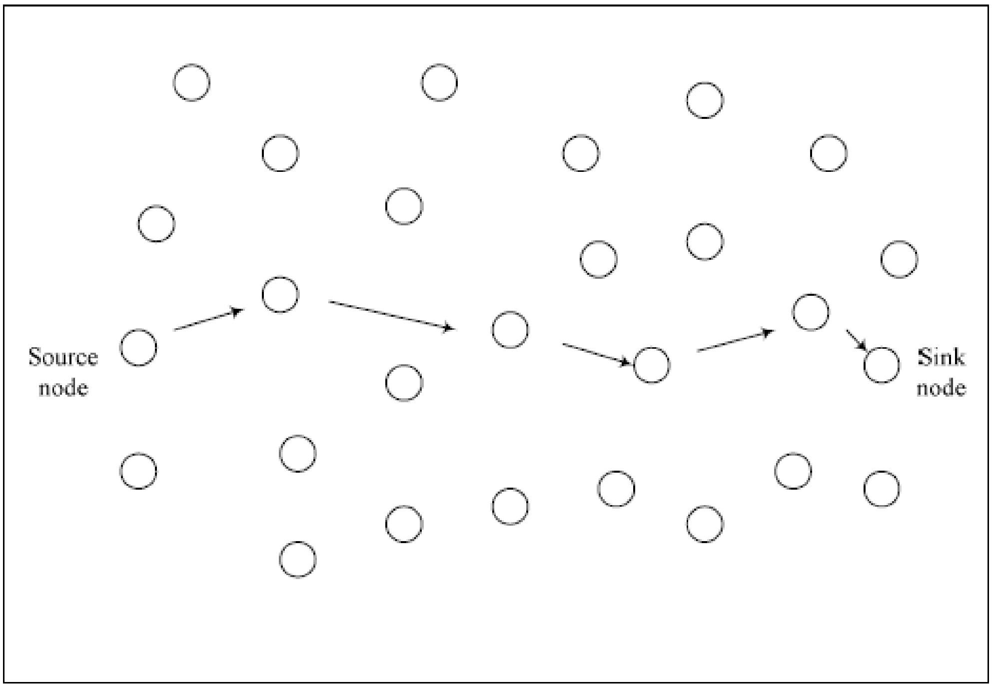


Figure 4. 2 a path from source node to sink node is discovered (He 2010).

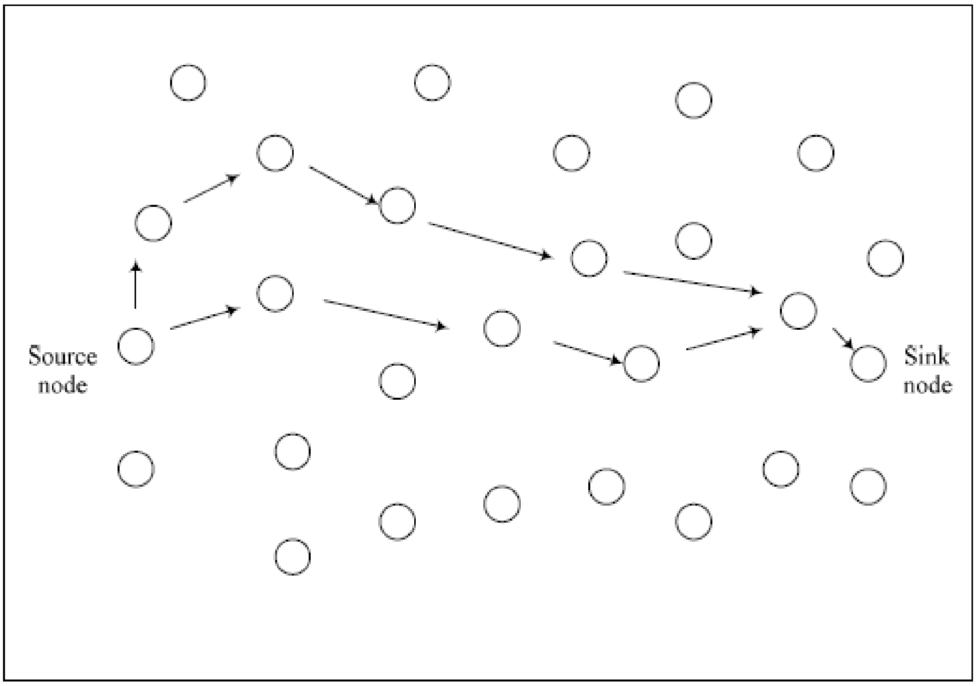


Figure 4. 3 another path from source node to sink node is constructed (He 2010).

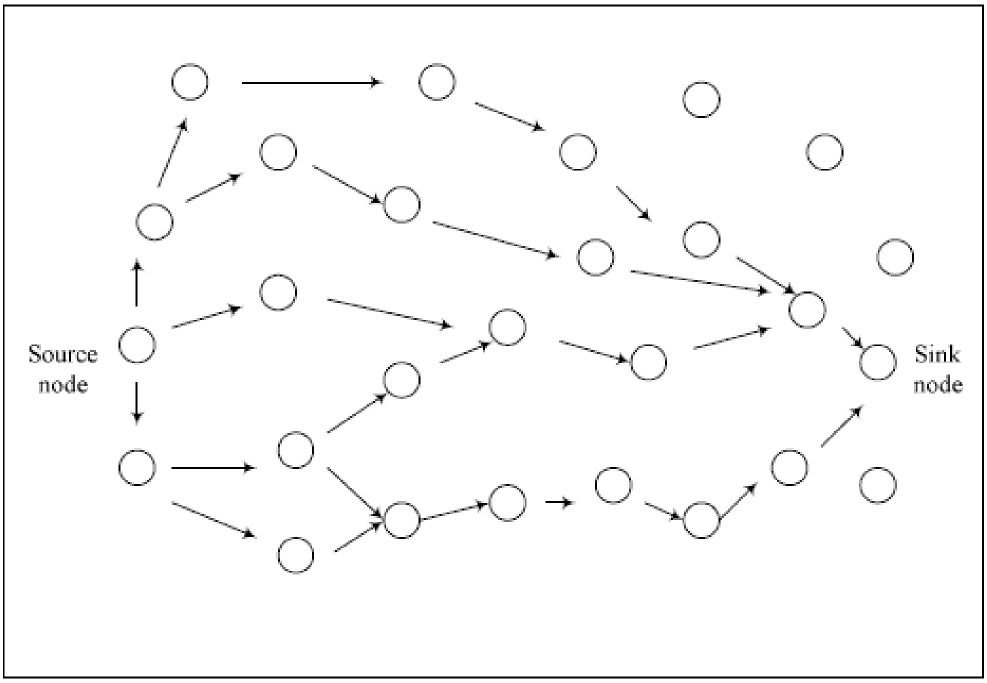


Figure 4. 4 multi-path routing (He 2010).

4.3.2.3 Query-Based Routing

Query-based routing is reactive. In query-based routing, there is a two phase process for route discovery, namely: the query phase and the route reply phase. At route discovery stage, a destination node propagates a query for data from a node throughout the entire network. This is called the query phase. When a node receives a query message it will reply to the query provided it has a data matching the task and that the same query was not already received. In the event that it has no matching data and that it is a new query the node will simply forward the query. In order to do all these tasks efficiently a node may keep a routing table, queuing list table etc. The query messages usually have a field that keeps track of the hops it traverses and their Ids as well. This information is used to send the requested message back to the node that raised the query.

With queries flooded throughout the network there is a possibility for the packet being routed infinite number of times. This will undoubtedly deplete the network of its valuable energy and thereby partition the network. A simple mechanism to avoid such a situation is to have a time-to-live (TTL) field in each query packet. The TTL can be set to say 100, and each time the packet is forwarded by a node, the TTL can be decremented by 1. In the event that the TTL value reaches 0, then the packet is discarded so as to avoid the route to infinity problem (Pearlman and Haas 1999).

In (Chim 2005) the Along and Across algorithm is proposed and it consists of a 5phase process. The first phase is the building of the loop free phase which is randomly initiated by a root node by generating a packet called a Tree-Build packet with a format (Sender_ID, Hop), whereby SenderID is the root ID and the hop value is initialised by the root to 0. This Tree-Build packet is then broadcast by the root to all its neighbours. Once the neighbours receive the Tree-Build packet they continue the tree building process by updating and flooding the packets to the entire network. The nodes that are direct neighbours to the root receive the first Tree_Build packet. The nodes copy the value of the Hop in the Tree_Build packet locally, then increment the hop value in the Tree_Build packet and *replace the sender ID by their own ID* and broadcast the modified packet to all their neighbours. After the first time of receiving the Tree_Build packet each time a node receives a Tree_Build packet it will inspect the packet and compare the HOP value with its locally stored value. In the case that the locally stored value is greater than the received one, the node will update its value with the received Hop value, it then increments the Hop value in the Tree_Build packet and broadcasts the packet to its neighbours as usual. Otherwise, the node discards the packet and does not update its Hop value. With this process all the nodes will get their Hop value which is the number of hops to reach the sink. Neighbours can overhear, the Tree-Build packet transmitted by a node as a result of the broadcast nature of wireless channels. During the tree formation process, each node builds up a list of its direct neighbours together with their hop levels. In this way a node knows its upper or lower level neighbour nodes.

This stage is then followed by distribution of Events Attributes. For this to happen nodes form an event Table keeping all known events and their forwarding directions. A packet called Event_Dist packet with the format of EventAttributes, Event SrcLevel is created by the event source node, the EventSrcLevel contains the hop level of the event source from the sink. When this type of packet is created by a source node, the node inspects its neighbour list to see whether there are neighbours matching the targeted event level in the EventSrcLevel. If it has such neighbours it will forward the packet to them, otherwise it forwards it to its one Hop upper level neighbours or the one Hop down level neighbours. As the event packets are propagated across the network, any node that receives an Event_Dist packet inspects its event table to see if the event has already been received before. If it has been received, then the new packet is discarded. Otherwise, the event attributes are recorded in the event Table and the forwarding direction of the particular event is set back to the sender and then the Event_Dist packet is forwarded using the set of rules discussed for the event packet. With this the Event_Dist message is transmitted across the network.

The next stage is the Basic Propagation of Queries. When a node wants to query for a particular event, it creates a Query_Prop packet with a format (QueriedEventAttributes, HopList) in which HopList records hops traversed by the query packet. The Query_Prop is randomly sent to either an upper or lower level node. When a node receives the Query_Prop, it then inspects it and updates the HopList field by appending its ID to it. After it inspects its Event Table for a match, if there is a match it then forwards the packet to the neighbour leading to the event. *If there is no match it forwards the packet to a neighbour in the opposite direction of the packet*. That is to say, if the packet was received from an *upper level neighbour*, then it forwards it to *a lower level neighbour* and vice versa. If the node is among the highest level node or the lowest level node then it cannot forward to higher level node or lower level node. In this case when a node receives a query it creates a packet called Query_Failed and transmits it back to the query source using the information on the HopList. When the query source receives the Query_Failed packet, it creates another Query_Prop packet and transmits it in the opposite direction to the one that resulted in a failure using the same forwarding scheme. In the event that both queries result in failure then the query is flooded to the entire network to look for a positive reply. The problem of an uppermost level node not being able to transmit to an upper level node was addressed by a scheme called Advanced Propagation of Queries.

The final stage is called the optimization return path which helps to use the shortest hop possible for transmitting the message. If an event source receives a Query_Prop packet that matches its events, it creates a Query_Answer packet and transmits it back to the query source with the help of the HopList information. The proposed optimization scheme makes it possible for a node forwarding the Query_Answer packet to inspect the HopList associated with the reply and forwards the *packet to the furthest neighbour thereby skipping some hops and shortening the hop count*.

This Along and Across algorithm was simulated in C++ and compared with Rumor Routing algorithm (Braginsky and Estrin 2002) and the results show that the Along and Across algorithm performs better than the Rumor Routing algorithm.

4.3.2.4 Location-Based Routing

This type of routing algorithm is used mostly for applications whereby the location of the phenomenon to be monitored is as important as the data to be transmitted. In such routing algorithms the sensor nodes are given addresses that reflect their positions or locations within the network. There are various schemes in obtaining the location of the sensor nodes. In (Bulusu et al. 2000; Capkun et al. 2001) the signal strength between neighbouring sensor nodes is used to estimate the distance between them. Once the distance between neighbouring sensors is estimated, the result is exchanged between neighbours and based on this the relative coordinates of the neighbouring sensors are derived. Another way of obtaining the location of the sensor nodes is by attaching small low power GPS receivers to them thereby getting their locations through GPS (Xu et al. 2001).

In Total GPS-free localization protocol for vehicular Ad Hoc and Sensor Networks (VASNET) a novel location protocol was proposed for vehicular Ad hoc and Sensor Networks (VASNET). However, the same scheme can be modified or even directly used in simple WSNs to get the location of sensor nodes. More importantly, the proposed scheme does not rely on GPS and therefore is free of some of the non-availability issues of GPS due to coverage and signal strength issues. This scheme consists of mobile nodes which are embedded in vehicles and stationary nodes which are sensor nodes deployed on either side of a highway. These sensors are called the Road Side Sensors (RSS), and they are equipped with the knowledge of their location coordinates. For a moving vehicle along a highway to know its location or position, its embedded sensor sends a packet called Location Request Packet (LREQ) to any of its one-hop neighbours be it RSS or mobile nodes. An RSS replies to LREQ with a packet called Location Reply Packet (LREP) which contains the coordinates to the LREQ sender which is the vehicle. In order for the vehicle to know its location it needs at least three LREP replies from different nodes. It then calculates the distance (L) between itself to the responding nodes with the help of the formula given below:

$$L = \frac{x_1}{\sqrt[n]{\frac{p_r}{p_s - x_z}}} \qquad (4.7)$$

Where x_1 is the carrier's wavelength, x_2 is the antennae gain, p_s and p_r are the power levels of the sender and receiver respectively and n is a physical environment constant. The calculated distance or length between the vehicle and the RSS can be used in the following triangulation formula to obtain the coordinates of the vehicle.

$$(x - x_A) + (y - y_A) = l_{A-v} \qquad (4.8)$$

Where $l_{A\text{-}v}$ is the length between node A and the vehicle, and (x_A, y_A) are RSS_A coordinates. The same naming scheme applies for equations 4.9 and 4.10.

$$(x - x_B) + (y - y_B) = l_{B-v} \qquad (4.9)$$

$$(x - x_C) + (y - y_C) = l_{C-v} \qquad (4.10)$$

In situations whereby the vehicle does not receive the three needed replies especially when some RSS are out of service then the vehicle asks the RSS that reply to rebroadcast its packet so that other RSS can reply. With this scheme the position of a vehicle can always be obtained without relying on a GPS.

4.4 Summary

In this chapter the two routing scenarios sink-to-sensor communications and sensor-to-sensor communication were discussed. This was then followed by a discussion of the two main categories of routing protocols based on network structure and protocol operation. The network structure category is further sub-divided into: flat-based Routing, Hierarchical-based routing and Adaptive-based Routing. The protocol operation category is also sub-divided into Negotiation-based Routing, Multipath-based Routing, Query-based routing and Locationbased Routing. Examples of these protocols were also given.

Chapter 5

THE PROPOSE ROUTING PROTOCOLS AND THEIR STOCHASTIC ANALYSIS

Chapter 5

THE PROPOSE ROUTING PROTOCOLS AND THEIR STOCHASTIC ANALYSIS

5.1 Introduction

Substantial research effort has gone into the development of sensor nodes and routing algorithms for WSNs. A number of routing algorithms have been proposed (Akyildiz et al. 2002; Cordeiro and Agrawal 2006). A typical sensor node has the following basic components: a sensing module, a processing module, a transceiver, and a power supply. These tiny sensor nodes are densely deployed and are used in various applications such as environmental monitoring, target tracking, military, habitat sensing, danger alarm and medical analysis (Akyildiz et al. 2002). One of the key issues in WSNs is the data delivery scheme between sensors and the data collection unit called the sink. The sink serves as the gateway between the network and the end users. The sink is a special node that is reliably connected by wire or satellite to the Internet and has adequate power supplies and processing power. All the other sensor nodes are so tiny that their energy supply, storage space, data processing and communication bandwidth are very limited, and therefore every possible means of efficient usage of these resources is aggressively sought. In WSNs the energy of the node is an invaluable commodity and hence various schemes such as the design principles for the network topology, routing algorithms or MAC protocols have been manipulated to maximize the network lifetime (Vergados et al. 2008; Beraldi et al. 2009; Guan et al. 2009). The sources of energy consumption are computation and communication. The communication between nodes is the major source of energy consumption in WSNs and hence is the most expensive aspect in conserving the energy. In this thesis, novel techniques for routing algorithm are proposed for the maximization of the network life time.

The size of the message routed in WSNs is application dependent. For this reason sensed data are classified according to their size namely: large-size, medium-size and small size data. BRALB is intended for applications that use small size data such as environmental monitoring. In such applications the message to be sent is comparable in size to the inquiry message between the neighbouring nodes. A WSN consists of a large number of nodes that are densely deployed in random fashion and as a result their design and maintenance is a challenging problem. For the sensors to adequately monitor some physical quantity in a given area, the sensors need to cover the area without leaving any void or un-sensed area. Hence, there is a need for deploying large number of nodes as the sensing range of each node is limited and the nodes are randomly deployed. Because of the large number of nodes deployed and the limited energy and computational power available to the nodes in WSNs, it is prohibitive to use routing protocols that require global information on the whole network.

In this thesis a routing protocol based on random walk for a square grid network topology has been proposed. This routing algorithm *does not require any global or location information* and *possesses an inherent load balancing property* for WSNs, which is difficult to achieve with any other routing protocol. In BRALB the probability of successful transmissions from the source to the destination by random walk using the proposed protocol has been *analysed statistically,* and it is proved that *random walk is as energy efficient as the shortest path routing algorithm while avoiding network partitioning due to repeated use of same routes in the network* (Touray and Johnson, 6th June 2012a). The qualifying assumption is that the message to be sent is small in size compared to the inquiry message among the neighbour nodes. BRALB uses probability to enable the nodes to make a fair guess about the energy level of their neighbouring nodes and selects the node with the maximum energy to forward the message (Touray and Johnson 2012a).

5.2 Related Works

In recent years, WSNs have been attracting rapidly increasing research efforts (Jiang et al. 1998;Meguerdichian et al. 2001; Cordeiro and Agrawal 2006). Research into routing protocols has been the major highlight of WSN research. The traditional routing schemes proved unsuitable to adapt to WSNs due to the specific characteristics of WSNs and hence several new algorithms (Tian et al. 2005;Guan et al. 2009;Ye et al. 2001;Shen et al. 2008;Guo et al. 2010;Ghane and Rajabzadeh 2010;Wei and Zhi 2010) that can support WSNs have been developed instead.

The idea for the family of ant algorithms stemmed from the behaviour of ants which communicate with each other by the use of a chemical substance called pheromone. The ants release pheromone on their path and these influence other ants to follow the same path. At the beginning there is no pheromone on the branches and ants have no clue about the length of branches. Also, in the beginning there might be several paths leading to the destination, but as time goes by the shortest path will have more pheromone. The more the pheromone laid by the ants on a path the larger the probability that they will visit the path next time. Thus there is a positive feedback in the groups of ants.

Roth and Wicker (2003) present an algorithm that closely relates to simple ant routing algorithm (ARA) by using the same distance vector principles with some difference in terms of route discovery and failure recovery. This algorithm requires the use of hello messages to initialise the links between neighbours. Once this is done, the hello packets are never sent again unless when a node's routing table is empty. In the Termite algorithm (Roth and Wicker 2003), a node discovers a route by sending an RREQ packet. This message walks in a random fashion and the subsequent hops being created or deleted in each stage before reaching its destination is done based on a uniformly distributed random decision function. When an RREQ packet finds a node that knows how to get to the target destination, the node generates an RREP packet and sends it back to the source. In this way packets are sent from source to destination. The RREP packet follows the pheromone laid by the RREQ, and further strengthening the pheromone along that path, thereby creating a clear path for the data packet to take.

In (Ghane and Rajabzadeh 2010) an energy efficient routing protocol called Remaining-Energy Based Routing (REB-R) is proposed and implemented. Its performance has been compared to the two well-known protocols AODV and T-ANT using NS-2. The main idea of REB-R is to broadcast the nodes' remaining energy along with the data in the data packet. In order to do this, two short packet types are created namely FWD_ROUTE and DATA. Shortly after the deployment of the sensor nodes the sink broadcasts FWD_ROUTE packet to all its neighbours. A node that received this packet stores it as FIRST_TTL in its TTL (time-to-live). This node will then broadcast a fresh new FWD_ROUTE packet with the value of TTL incremented by one. Nodes use TTL value to form neighbours, this value also shows how far a node is from the sink. Level-one nodes are neighbours to the sink while level-two nodes are neighbours to level-one nodes and so on. When a node received a FWD_ROUTE packet with a smaller TTL or equal to its FIRST_TTL it is recorded as *nominee* for the parent node of the receiving node. If the TTL is more than FIRST_TTL then the packet is dropped to avoid broadcasting unnecessary information. During this period the sender of any packet that has minimum TTL and maximum energy is selected as the current parent node. When the network has reached a steady state, current parent node is selected from the list of nominees which has the maximum energy. After this the nodes will have enough information to route packets to the sink. Though this algorithm performs better than AODV and T-ANT, it has lot of overheads and might deplete the nodes' energy more rapidly compared to BRALB.

In (Wei and Zhi 2010) an advanced ant colony algorithm based on cloud model (AACOCM) which is based on ant colony optimization algorithm (ACOA) is proposed. AACOCM is a multi-objective bandwidth constrained algorithm and it aims to find a route in the network which has sufficient resources to optimize some of the network parameters such as data packet lost rate, delay and energy consumption. This algorithm introduces a multi-objective evaluation function by considering the three mentioned parameters. Three ant types are defined namely: ordinary ants, greedy ants and unusual ants with different behaviors which help to achieve the multi-objective inspection and evaluation. The ants walk from the source to the destination in a gradual routing tree formation.

In (Guo et al. 2010), a routing protocol for WSNs based on the Power-Efficient Gathering in Sensor Information Systems (PEGASIS) protocol using an improved ant colony algorithm to construct the chain, instead of the greedy algorithm, is proposed. This protocol is called PEG-ant. In the chain construction process, in order to choose a node as the next one on the chain, all the current node's neighbours are candidates whereby the selection criteria are the remaining energy of the candidate, the amount of energy consumed when a unit data is transmitted along the branch between the current node and the candidate, and the pheromone quantity in a branch. It forms a chain that makes the path more evenly-distributed when compared to PEGASIS. A leader is selected for each round of transmission to directly communicate with the BS based on node with maximum remaining energy. Along the built chain and in the direction of leader, starting from the furthest nodes, each node fuses the received data with its own, *and and then ttransmit it as one packte to the other neighbours*. This will eventually reach the leader. The leader then sends the final fused data to the BS.

In (Baras and Mehta 2003), the proposed PERA algorithm uses Ant-like agents to discover and maintain paths in MANETs. At the initial stage, there are no routing tables or no next hop from source to destination. The initialization and neighbour discovery are done by single-hop, broadcast HELLO messages that are transmitted periodically in order to build the neighbour list. During the initial stages the first forward ant is sent by a node with equal probability to one of its neighbours. If a node has N neighbours the probability of selecting each neighbour as the next hop is $\mathbf{1/N}$. These probabilities will change from being uniform as they are adjusted when the source received the backward ants from the destination. Therefore a node will have different probabilities for forwarding to the next hop neighbour over time due to the backward ants' feedback.

Each neighbour has a routing table in the format [Destination, Next hop, Probability]. In addition, each node also maintains a table of statistics containing the *mean and variance* for each destination 'd' to which a forward ant has been previously sent. These routing tables are built and maintained by the information gathered from the Forward ants, Uniform ants, Regular ants and the Backward ants.

The forward ants are agents generated periodically by a node and sent to randomly selected destinations so as to gather routing information. A node that sends a forward ant creates a routing table if there is none for that node; the intermediate nodes also do the same thing. A forward ant packet contains: source, next hop and destination IP addresses, Stack and hop count. The Stack of the forward ants is a dynamically growing data structure that lists the IP addresses of the traversed nodes as well as the time. Forward ants face the same network conditions as they are routed just like normal data packets. The forwarding of the forward ant is probabilistic and provides exploration of paths available in the network. These ants were then called Regular Ants so as to distinguish them from Forward Uniform Ants. A node that receives a forward ant checks to see if it has previously traversed the node. If it has not, then the IP address of the node and the current time are pushed into the stack of the ant. On the other hand if it has and the node's IP address is found in the ant's existing stack, this implies that the forward ant has gone into a loop and is therefore destroyed.

In order to promote the discovery of new routes the authors in (Baras and Mehta 2003) created a uniform ant. These ants are created in the same manner like regular ants but are routed differently. The routing of the uniform ant does not use the routing table; instead it uses equal probability to select the next hop node. These ants help to explore and reinforce newly discovered routes and prevent the saturation of previously discovered paths. A backward ant is created when a forward ant reaches its destination. The backward ant inherits the Stack of the forward ant; it uses this to quickly update the source node and all the intermediate nodes along the path by using the high priority queue. PERA was compared to AODV, and the results indicate that it performs better than AODV in terms of delay.

The use of random walk in WSNs has been extensively researched (Shokrzadeh et al. 2007), of which the Directional Rumor Routing in Wireless Sensor Networks is based on the random walk of agents. The aim of this algorithm is to improve the latency and energy consumption of the traditional algorithms using propagation of query and event agents in straight lines, instead of using purely random walk paths. Directed Rumor Routing has two phases for calibration. In the first phase each node sends a Hello message stating its position to each of its neighbour. The hello messages are used by the receiving nodes to record their neighbours and their positions. During the second phase each node tests whether it is at the edge of the network. When a node senses an event, it creates a number of event agents and propagates them into the network along some linear paths forming star-like propagation trajectories. These event agents are not allowed to pass the edge nodes. After this a node is randomly selected as the sink node. The sink node creates some query agents for each fired event. Each agent contains the id of the current node, the id of the previous node (depicting the direction of events), location information of the source node, and a table containing the ids of the events and distances to them. The disadvantage with this method is that the hello messages drain the WSN of its limited bandwidth and imposes additional energy drain on the nodes.

However, *in BRALB the need for hello messages has been eliminated thus reducing the energy consumption during the routing process*.

Sensing and connectivity are vital parts of WSN. For efficiency it is important to use a minimum number of sensors possible while maintaining the connectivity to cover a sensing area in order to reduce the cost. The connectivity requirement is to ensure that all the nodes are able to communicate with the sink either through single or multi-hop communication, while the sensing requirement is to make sure that the entire WSN is at least within the sensing area (Tian et al. 2005).

Various topologies have been proposed to address both coverage and connectivity issues. If the region to be covered is in 2-dimensional plane three types of regular topologies are used for WSNs namely: square, hexagon, and triangle-based topologies. It has been demonstrated in (Tian et al. 2005) that among those three topologies, triangle-based topology provides the best sensing strength and reliability while trading off energy consumption and total coverage. While WSNs in hexagon-based topologies provide maximal connected-coverage given the same number of nodes, Square-based topology's performance lies between the two and yet is the simplest architecture. It is for this reason that the analysis of BRALB is based *on a square grid type network topology*.

5.3 Biased Random Algorithm for Load Balancing (BRALB)

BRALB is implemented on a square grid type network topology with each node having four neighbours except for the border nodes.

In BRALB a node will have a message counter for each neighbour. When node A sends a message to node B, the node A will increase the counter designated for node B by a value of 1. On the other hand the receiving node B will update its counter designated for node A; increasing it by a value of 4. The reason for the four step increment for a message received is to try and predict the number of messages sent by that particular neighbour. It is a known fact in statistical studies that when one node has an equal chance of selecting four neighbours to send messages, if one of those neighbours received say 5 messages then it is reasonable to assume that the remaining three neighbours may as well have received 5 messages each. In this way it is a fair guess to say that the sending node must have sent 20 messages in all. By the same argument for every message received from a neighbour one can safely assume that the neighbour may have sent 4 messages, thus justifying the assumption of a four step increment. In this way all the messages sent and received by a neighbour can be predicted and therefore the energy level of that particular node can be predicted as there is a correlation between node energy and messages processed by that node. There could be many sources sending messages and this assumption then may be very significant but in WSNs there are large numbers of sensors and hence many activities going on. Therefore, when a node start sending a message, it is very likely that it will send more messages over time making the assumption justifiable. On the worst case scenario even if many nodes are just sending few messages, then the assumption will be made to all of them and that will in effect cancel out. Therefore, when node A with four neighbours wants to send a message to the sink which is say six hops away from itself, it will first inspect its counters and select the node with the least count. The counters are all initialise to zero and are incremented in steps of 1s or 4s for sending a message or receiving a message respectively .This process is repeated until the message reaches the Sink. In BRALB it has been proved statistically that in application whereby the message to be sent to the Sink is comparable to the size of the inquiry message then this routing mechanism can route to the Sink by using the same energy as the SPF algorithm.

The messages sent and received represent the energy used in the network and therefore by biasing the nodes to *forward to the nodes with the lowest message counter* (both sent and received) *the network load balances the energy of the network*. This will avoid using the shortest path all the time by distributing the energy usage fairly within the network and hence will avoid partitioning of the network.

5.4 Statistical Analysis of BRALB

BRALB can be statistically analysed on a square grid based network topology by investigating the probabilities of forwarding a packet through the square grid network topology using the algorithm.

When a node wants to report an event to the sink or base station, in WSNs, it would usually contact all its neighbours and then forward it to the neighbour with the least number of hops to the sink. *In applications where the message to be sent is small* in size such as abnormal activity detection system and danger alarm system, *the communication cost between neighbour pairs for choosing the next-hop neighbour is comparable with that of transmitting the real message itself* (Tian et al. 2005; Touray, Johnson et al. 2011). In such traditional WSNs topologies, a node needs to *communicate with three of its neighbours* before forwarding the message. A forwarding node, will not contact the node it received the message from, therefore it has three neighbours to contact. A source node, on the other hand will need to contact all of its four neighbours. As the communication cost between two nodes for next-hop inquiry can be safely assumed to be equal to the transmission cost of the real message to be sent, it can be denoted by one energy unit ‘e’. When a source node wants to send a packet to the sink, it will contact its four neighbours before selecting the next hops. The energy used by the source to make this possible is therefore four units of energy plus the extra unit for sending the actual message.

For the next-hop nodes, the neighbour nodes will send a response to the sender to confirm that they have a valid route to the destination after being contacted. The message is then transmitted to this node which then forwards it to its chosen neighbour.

Let us assumed that the cost of transmitting data from one hop to another, is one unit of energy. Therefore the total energy for routing data from one next-hop node to its next-hop node is the sum of energy units required for responding to route request (1e), the number of energy units required for contacting its three neighbours (3e), and the energy units required for transmitting the real data (1e), which totals to 5e.

However, the cost of routing data to the next hop in random walk is just one unit of energy 'e'. This is because in BRALB the *data is just forwarded base on local knowledge of the node to any of its three neighbours without any initial inquiry*. In Fig. 5.1, if data is to be sent from a source say G to the sink Y which is six hops away then the energy cost would be 30e for the shortest path routing and only 6e for BRALB if it routes the message itself in 6 hops.

In BRALB when a node receives a packet, the next hop is determined by the number of packets it had already received or sent to its neighbours. Based on this it will select one neighbour and forward the packet to *the neighbour who has either sent or received the least number of messages*. Therefore, the energy cost for one hop transmission using the proposed algorithm is only one unit of energy 'e', while the energy cost for one hop of data transmission using the traditional shortest path is five units of energy (5e), including the inquiry stage. In general, the shortest path routing protocol uses the least amount of energy in routing a message from source to the destination. The effectiveness of BRALB in terms of energy usage is proved to be better than that of the shortest path routing protocol under certain conditions. The probability of successfully sending data from the source to the sink will be analysed in order to determine the effectiveness of BRALB when used in a squared grid type network topology.

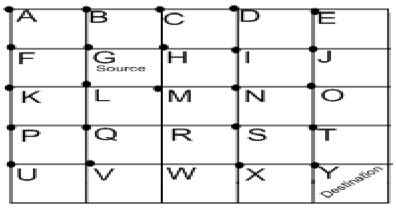


Figure 5. 1 A square grid-based WSN topology

Figure 5.1 illustrates an example of sensor network consisting of 25 nodes named from A to Y arranged in a square grid type network topology. In this network, node G is designated as the source and node Y acts as the sink/destination. The routing scenario where a node cannot forward a message to the node it received the packet from will be assumed. So, only the source node will have four neighbours to select from, and the rest of the nodes along the path will have only three neighbours to choose from as long as they are not border nodes. Therefore it is fair to assume that on an average all nodes will have three neighbours to select from.

In a square grid type network topology, the expression for the relationship between the boundary nodes and the total number of nodes can be developed as given below. Let the total number of nodes in the grid be represented as $\boldsymbol{m} * \boldsymbol{m}$ and the number of boundary nodes be represented by b. Then the total number of boundary nodes in any given square grid can be written as:

$$\mathrm{b} = \mathrm{m} + \mathrm{m} + (\mathrm{m} - 2) + (\mathrm{m} - 2)$$

$$\mathrm{b} = 4\mathrm{m} - 4 \tag{5.1}$$

Then the ratio between the boundary nodes and the total number of nodes is given by,

$$\frac{4\mathrm{m}-4}{\mathrm{m}*\mathrm{m}} \tag{5.2}$$

The decimal value 4 in the numerator of equation 5.2 can be ignored as $m \to \infty$

$$\lim_{m\to\infty} \frac{4m}{m^2} = \frac{4}{m} \to \tag{5.3}$$

Hence it is reasonable to assume that the non-boundary nodes are negligible as the grid size becomes larger.

In Figure 5.1 if node G sends a packet to either F or B it is in the wrong direction, whereas if it sends to either H or L it is in the correct direction towards the sink Y. Using this condition we can calculate the probability of a data packet being successfully sent from node G to the sink node Y. The sink node Y is six hops away from the source node G.

In this scenario it will be assumed that all the neighbours have equal chances of sending or receiving from a neighbour. Therefore the chances of selecting any of the three neighbours are the same. The probability that a packet is forwarded in the wrong or correct direction *depends on whether a forwarding node has received the packet from the wrong or correct direction*.

For the packet to be successfully routed from node G to node Y, using the shortest path over six hops, it must be forwarded in the right direction for all the hops. The probability that this happens is calculated as follows. If a node received a packet from a correct direction then the probability of forwarding in the correct direction is calculated as follows. A node has three neighbours to select as next-hop. Two of these neighbours are in the correct direction and have a probability of $\frac{2}{3}$ for forwarding in the correct direction while the remaining neighbour is in the wrong direction and has a probability of $\frac{1}{3}$ for forwarding in the correct direction. At the initial stage when a neighbour has three messages to send, BRALB will first pick one of its neighbours at random and will forward the first message. For the second message it will pick one of the other two neighbours as the next-hop while the last message is sent to the remaining neighbour. This process is repeated for every three messages a node has to send. Hence the total probability of forwarding a packet in the correct direction when it has been received from the correct direction is the average of the probabilities of the three neighbours viz $\frac{1}{3}.\frac{2}{3}+\frac{1}{3}.\frac{2}{3}+\frac{1}{3}.\frac{1}{3}=\frac{5}{9}$. A packet is said to be received in the wrong direction if it is sent further away from the destination. *If the packet is received from the wrong direction then two of its neighbours are in the wrong direction and only one neighbour is in the correct direction* (i.e. in the same direction as the sink). The two neighbours in the wrong direction have a probability of $\frac{1}{3}$ each of forwarding in the correct direction while the remaining neighbour is in the correct direction and has a probability of $\frac{2}{3}$ forwarding to the correct direction. Hence the total probability of forwarding a packet that's received from the wrong direction to the correct direction is $\frac{1}{3}.\frac{1}{3}+\frac{1}{3}.\frac{1}{3}+\frac{1}{3}.\frac{2}{3}=\frac{4}{9}$. If a node received a packet from a correct direction then the probability of forwarding in the correct direction is $\frac{5}{9}$ and the probability of forwarding in the wrong direction is $\frac{4}{9}$. On the other hand if a node received a packet from a wrong direction then the probability of forwarding in the correct direction is $\frac{4}{9}$ and the probability of forwarding in the wrong direction is $\frac{5}{9}$.

If the packet is to be forwarded in six hops to reach the destination along the shortest path then it must be forwarded along the correct path in six hops and along the wrong path in zero hops. The probability of this happening is given by the equation below:

$$p\{d=6\}=\binom{6}{0}\frac{5^6}{9^6}=0.029401 \qquad (5.4)$$

Where p is the probability that a packet is forwarded and d is total number of hops the packet travels.

However, it is very likely that the packet will not be forwarded through the shortest route by random walk. Therefore it is likely that the packet will be forwarded in the wrong direction before getting to the destination. If the packet is sent one hop in the wrong direction, it must move one hop backward towards the correct direction. Therefore, for every one hop in the wrong direction two hops must be added to ‘d’ to get the total number of hops to route the packet to the destination. In this case where the destination is six hops away from the source, if the packet is forwarded one hop in the wrong direction, then the least number of hops to send it back to the destination would be 8. Therefore, the relationship between the total numbers of hops a packet traverses and the number of hops it traverses in the wrong direction is given as:

$$d = k + 2.i, \tag{5.5}$$

Where d is the total number of hops a packet travels, k the smallest number of hops between the source and destination and ‘i’ the number of hops the packet is forwarded in the wrong direction. In the above example, k = 6, d = 6 + 2.i., and ‘i’ must be zero and therefore if the total number of hops (d) is six, the packet was forwarded in the correct direction all the time. With the total number of hops being 8, the packet would have been forwarded 7 hops in the correct direction and one hop in the wrong direction. Similarly for d=10, the packet would have been forwarded 8 hops in the correct direction and two hops in the wrong direction. This can be used to easily calculate the probability of successful transmissions for any total number of hops. The probability of successful transmissions at 12 hops is shown below:

$$p\{d = 12\} = \binom{12}{3}\left(\frac{9}{12}.\frac{5}{9} + \frac{3}{12}.\frac{4}{9}\right)^{9}.\left(\frac{9}{12}.\frac{4}{9} + \frac{3}{12}.\frac{5}{9}\right)^{3} \tag{5.6}$$

Therefore the probability of a packet reaching the destination at any given number of hops d in scenario 1 can be written as:

$$p\{d = k + 2i\} = \binom{k+2i}{i}\left(\frac{k+i}{k+2i}.\frac{5}{9} + \frac{i}{k+2i}.\frac{4}{9}\right)^{k+i}.\left(\frac{k+i}{k+2i}.\frac{4}{9} + \frac{i}{k+2i}.\frac{5}{9}\right)^{i} \tag{5.7}$$

$$p\{d \leq H\} = \sum_{i=0}^{\frac{H-k}{2}}\binom{k+2i}{i}\left(\frac{k+i}{k+2i}.\frac{5}{9} + \frac{i}{k+2i}.\frac{4}{9}\right)^{k+i}.\left(\frac{k+i}{k+2i}.\frac{4}{9} + \frac{i}{k+2i}.\frac{5}{9}\right)^{i} \tag{5.8}$$

Hence the probabilities for successful transmission with 6, 24, 30 and 40 hops are calculated by substituting these values for H and substituting k by 6 as in our example as follows:

$$p\{d \leq 6\} = 0.02940119411 \quad (5.9)$$

$$p\{d \leq 24\} = 0.7401902058 \quad (5.10)$$

$$p\{d \leq 30\} = 1.014642274 \quad (5.11)$$

$$p\{d \leq 40\} = 1.464255858 \quad (5.12)$$

In equation 5.12 it can be seen that the successful transmission probability was more than 100%, this is the result of assuming all the nodes were non-boundary nodes. Hence it can be deduced from the above calculation that, using the proposed biased random algorithm BRALB, it has been proved that it guarantees 100% of the packets to be successfully transmitted using the same energy (30e) as the SPF algorithm.

The SPF routing requires 5k energy cost as calculated above. In (Tian et al. 2005) it is found that flooding algorithm requires 2m (m – 1) energy cost in a WSN with a grid size of m * m, m being the number of nodes in each edge. Table 5.1 compares the energy cost of these algorithms by routing messages along the longest route in the grid (k = 2(m-1)).

Table 5. 1 Performance comparison of SPF, flooding and BRALB

Routing Algorithm	Number of hops	Energy cost
Shortest path	k	5k
BRALB	5k = (5*6) with 100% successful transmission rate	$\leq 5k$
Flooding	K(k/2-1)	K(k/2-1)

5.5 The new Proposed Cluster Based Routing Scheme C-BRALB

An improved routing scheme is proposed which is based on BRALB routing protocol with a slight modification and introduction of clustering mechanism for an additional layer of randomness. In C-BRALB when a node has a data to transmit it sends it to its cluster head. It is the duty of the cluster head to aggregate traffic and to transmit it to the base station via the neighbouring cluster heads. Two Cluster Head nodes (CHNs) are neighbours when they can communicate with each other. There is always one CHN in a cluster at a time. A CHN keeps a counter for each neighbour CHN where it records the number of messages it had sent or received from them. When a CHN wants to forward data, it inspects its neighbour CHN counters and then forwards the data to the CHN whose counter has the least number of messages. The messages sent or received by a CHN can be used to estimate the energy value of a cluster. Therefore the CHN can always route to the neighbour CHN with more energy by inspecting its CHN counters and hence load balance the traffic across the network. The clustering scheme proposed is based on *the IDD clustering technique* with *some modifications*. The major differences are the omission of Interest Diffusion phase and the introduction of a totally different Data Propagation phase. C-BRALB is composed of *two phases* which are the *Cluster Formation* and *Data Propagation phases*, whereas the IDD clustering technique constitutes five phases.

The first phase of C-BRALB is the *cluster formation protocol* which is made of two phases namely initial phase and cluster formation phase. A node can exist in the following states namely: ordinary nodes (ORN), cluster head node (CHN), cluster member node (CMN), candidate cluster head node (CCHN), and finally border node (BN). All nodes start as ordinary nodes and they may transition into other states after the cluster formation protocol process.

The cluster formation protocol in C-BRALB is based on energy threshold TH_e propagated by the base station to the entire network. A node that receives this broadcast message will inspect the message and compare its residual energy to the energy threshold TH_e, setting. If the TH_e is greater than the node’s residual energy, then the node does not change its state. On the other hand if the node's residual energy is not less than TH_e, the node transits to the CCHN state, and then declares its new state to the neighbours. In case of two or more neighbour nodes being in the CCHN state at the same time the tie breaker is that the neighbour with the highest residual energy will win and transit to the CHN. The winner then broadcasts its CHN state to its neighbours. The CCHN nodes that lost the tie breaker will then transit to ORN state. The ORN nodes that receive a cluster head node message will join the cluster and set their state to CMN. The ORN nodes that receive two or more cluster head node messages at the same time will set their node state to BN. Each cluster head will also have an energy threshold called TH_c which is used to initiate cluster head rotation. When a CHN node's energy falls below TH_c it then relinquish its cluster head duties to another node within the cluster.

In order to maintain the clusters, a CHN keeps a member information table to store the information of each CMN in the cluster and a message counter for each neighbour cluster. When the cluster is fully formed, the CMN nodes put their Id information, energy level information and position into a message called *mem_msg* and send it to their CHN node. It uses its member node table to record information about nodes within its cluster while the neighbour cluster counter is used to record the number of messages sent or received from neighbour clusters. When a CHN node receives a mem_msg it will check it against its member information table. If the record is already in the table the message will still be inspected for energy level update otherwise it is added to the table together with its new parameters. When a cluster head node uses its energy up to an energy threshold TH_c, it then broadcasts its TH_c value to all the nodes in its cluster. A node that receives this broadcast message will respond to the cluster head with a mem_msg updating its energy level to the cluster head. The cluster head will then inspect all the mem_msg packets and will select the node with the highest energy as the next cluster head. The previous cluster head will then update the new cluster head with the cluster member table and the neighbour cluster counters. This process is repeated every time a cluster head node's energy falls below the TH_c so as to rotate the cluster head duty among the nodes within a cluster. This helps to balance the transmission of traffic load among the nodes within a cluster.

The next phase is the Data Propagation. When a node has a data to send, it will transmit the data to the cluster head. When the cluster head receives the message it will send it to one of its neighbour cluster heads at random at the initial stage. After forwarding the message to a neighbour cluster it will then update the message counter for that neighbour cluster. If the same cluster head has a data to forward now it will inspect its neighbour cluster head counters and will forward to the one with the *lowest message counter*. In this way each cluster will load balance the data transmission to its neighbour clusters and hence maximise the network life time. This process is repeated every time a cluster head has a data to transmit. The cluster formation process for C-BRALB is highlighted in Fig. 5.2.

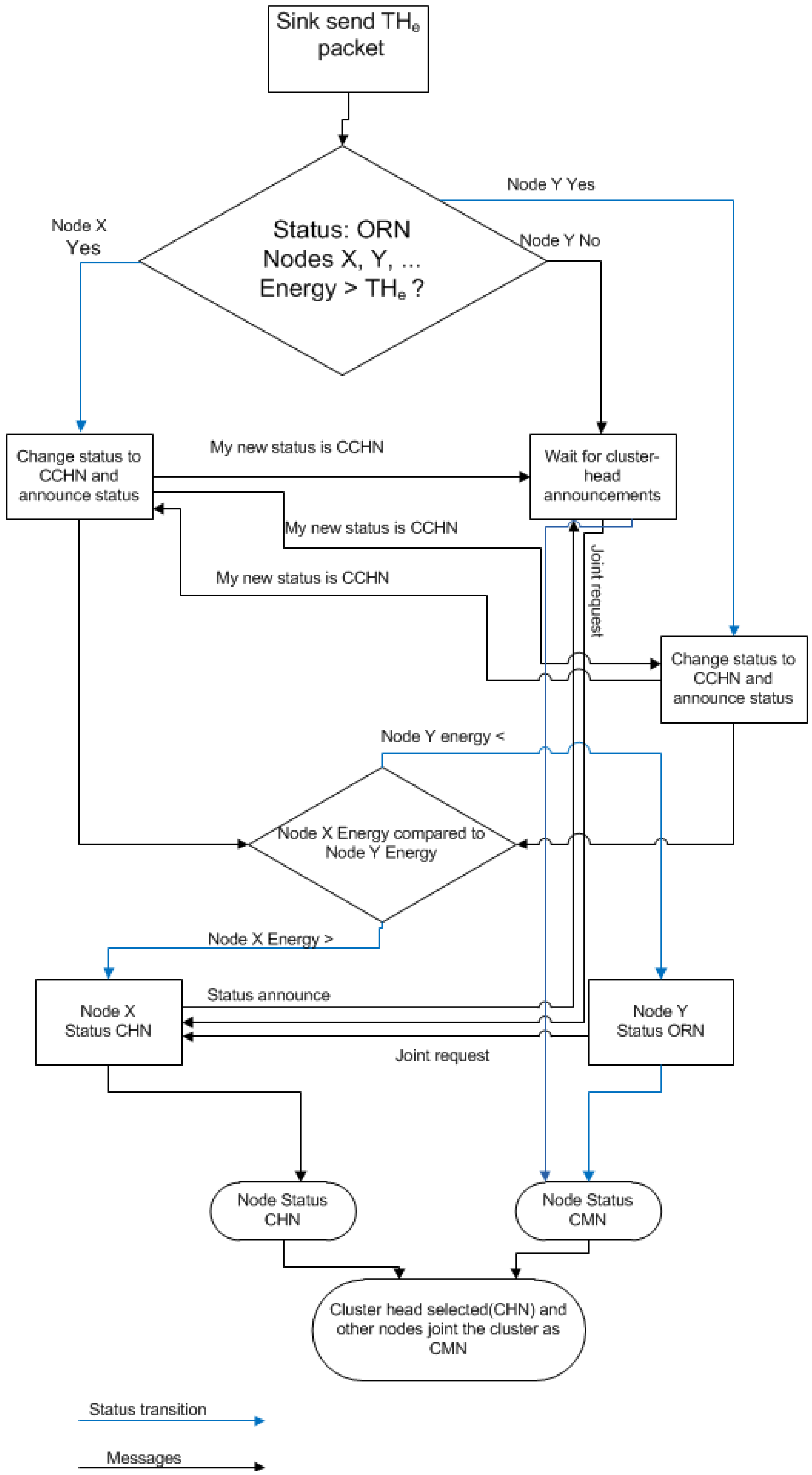


Figure 5. 2 Simple flowchart of the cluster formation of C-BRALB

5.6 Summary

In this chapter, BRALB a routing algorithm that gives a distributed load-balancing scheme by using a biased random walk in a square-grid WSNs is proposed. This routing algorithm is self-organised and depends only on local information for load distribution and route maintenance. This algorithm works on a square grid type topology and is suitable for applications whereby the data to be sent is small enough to be compared with the exchange information between neighbouring nodes. For these applications like environmental monitoring and danger alarm monitoring BRALB has been proved both statistically and empirically to consume at most the same amount of energy as the shortest path first routing algorithms. However, the problem with BRALB is that it is not scalable.

The scalability issue in BRALB is addressed by the introduction of a clustering scheme resulting in a new routing scheme called C-BRALB which is intended for both regular and randomly deployed networks. C-BRALB divides the sensor networks into many clusters and cluster heads are elected in each cluster. There is also a cluster head rotation within each cluster in order to make the algorithm more energy efficient. The cluster head is responsible for collecting and aggregating the sensed data within the cluster and to transmit it to the base station through a neighbour cluster head. Thus, C-BRALB can provide an effective load-balancing paradigm and is scalable for both grid and *randomly deployed* networks.

Chapter 6

SIMULATION AND ANALYSIS OF THE ROUTING MODELS

Chapter 6

SIMULATION AND ANALYSIS

6.1 Introduction

As mentioned earlier in this thesis, there are two types of approaches for modelling routing protocols namely: analytical and simulation approach. In the simulation approach a prototype of the protocol under consideration is built by factoring in all the relevant features of the protocol. BRALB and C-BRALB are simulated so as to evaluate their performance under different scenarios, topologies and configuration. In this section of the thesis, the simulation results will also be compared to the statistical analysis of the routing models.

In this chapter the proposed scheme is evaluated and compared to the two well-known protocols SPF and LEACH. In the literature review, it was highlighted that the best way to evaluate a new protocol is carrying out a fixed network evaluation and a scalability analysis. In the fixed network evaluation the proposed schemes will be evaluated and compared to SPF and LEACH while in the scalability analysis the proposed scheme will be independently analysed under various network settings.

6.2 Simulation of the Proposed BRALB Model

Netlogo simulator is used in order to evaluate and compare BRALB to some of the well-known existing routing algorithms. The environment used to test the performance of this algorithm was modelled using Netlogo's graphic design tool in order to simulate a network. By using Netlogo, the network parameters were varied in order to study their effect on the overall performance of each algorithm and to compare. The simulator facilitates to deploy the number of resource-constrained nodes and the required topology connectivity. The simulation was run on an m-dimensional network with $m \times m$ number of nodes each having four neighbours except for the boundary nodes for BRALB algorithm. In the simulation test bed as depicted in Figure 6.1, the total number of nodes was considered to be m × m with node connectivity of 4. Various network parameters were used to evaluate the performance of the two algorithms. The algorithms were implemented in the simulator for comparison. Tests were run for 100 time units, which were considered as ticks. For every performance metric, an average from 20 simulations were taken to generate each value.

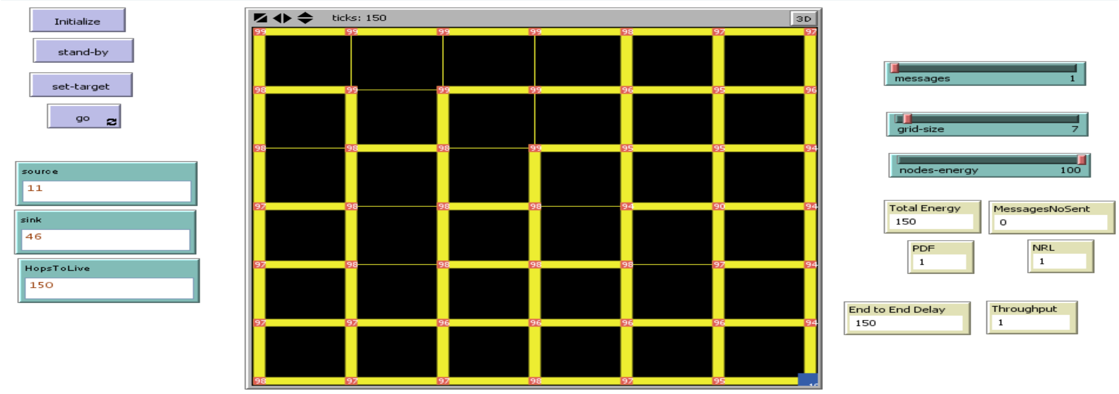


Figure 6. 1 Simulation snapshot for BRALB

6.2.1 Simulation Conditions for BRALB

In practical applications of WSNs, each node is powered by a battery with no mobility or very limited mobility, and hence, the topology changes only due to node failures. Therefore the following assumptions on the wireless sensor nodes and WSNs have been made for the proposed model architecture:

(1) Each node has the same limited initial energy except the sink which has unlimited energy; the links between nodes is symmetrical; nodes A and B can communicate with each other.

(2) All the individual nodes have the same computing capacity to support signal processing and routing.

(3) The nodes are all static and neighbours are within a node’s transmission range.

(4) Wireless signal communication cost (energy) is the same in all directions.

(5) The communication path between two nodes *is not affected by the interference* of other nodes (Luwei et al. 2011).

(6) The network is deployed in a two-dimensional field of finite area.

(7) Nodes are capable of measuring the signal strength of a received message (Hill et al. 2000).

(8) Each node is aware of its location.

(9) Nodes consume energy when transmitting, receiving and sensing, but not while idling.

(10) Underlying protocols make the nodes to be aware of the set of nodes within their transmission radius (Rachuri and Murthy 2011).

6.2.2 Simulation Setup for BRALB

The algorithm is assumed to use a simple radio model based on (Heinzelman 2000a) in which the transmitter, power receiver and the amplifier dissipate energy in order to operate the radio electronics. It is assumed in BRALB that the transmission energy required for one packet or byte is equivalent to receiving energy of one packet or byte. Therefore to transmit or receive a k-bit message, the energy consumed is 50 nJ/bit (Electronic Energy)* k bits (Heinzelman 2000a). Table 6.1 shows the simulation parameters for BRALB (Heinzelman 2000a).

Table 6. 1 Simulation Parameters for small coverage area of the cell

Parameter Values	
Initial Battery Level at an individual node level	5 KJ
Threshold Transmission distance in meters	100 meters
Transmission electronics Energy	50 nJ/bit
Sensed Message Packet Size by Sensor Nodes	600 bits (75bytes)
Energy to send or receive one 600 bits packet size message	50 nJ/bit*600 bits= 30µJ
Minimum Level of Battery Required for Node to remain functional	60 µJ

6.2.3 Simulation Results and Analysis for BRALB

The proposed routing algorithm BRALB is simulated and then first compared to the theoretical prediction using statistical analysis. It is then simulated and compared to the shortest path first (SPF) routing algorithm and then simulated alone for scalability analysis as suggested in the literature review.

The efficiency of the proposed routing algorithm has been measured on the basis of various parameters such as average energy dissipation, throughput, network lifetime, routing overhead, scalability and number of dead nodes.

6.2.3.1 BRALB Versus Theoretical Prediction

To study the performance of the proposed routing algorithm mechanism, the number of messages delivered in the simulation is examined and compared with the predicted statistical analysis. In the simulation, the *key factor considered for selecting the next hop* node is local knowledge estimated by the forwarding node about *the remaining energy values of its neighbours*.

Here, the performance of the proposed routing mechanism for wireless sensor network, where all nodes have equal capabilities and data packets can be sent by any node will be discussed. Such assumptions are made to prove the efficiency of the proposed routing scheme and to verify that the simulated results match the analytical results. This analysis is also helpful in carrying out the scalability analysis of BRALB as suggested in the literature review section of this thesis.

The simulation result is obtained by simulating transmission of one hundred messages to the sink within different grid size networks. The ratio of the number of messages sent in the network and the number of messages received by the sink is then compared to the theoretical prediction. In Figure 6.2 the message success rate is taken when the source and the sink are 6 hops apart. From the graph it can be clearly seen that the theoretical prediction agrees with the simulated results. From Figure 6.2, it can be noted that both the predicted and simulated results show one hundred percent success rate at 30 hops walk between the source and the destination.

Figure 6. 2 Message success ratio when a 5*5 grid WSN employs BRALB for routing

In Figure 6.3 the message success rate is taken when the source and the sink are 12 hops apart. From the graph it can be clearly seen that the theoretical prediction correlates to the simulated results. There is a hundred percent agreement between the simulation results and theoretical predictions at 80 hops.

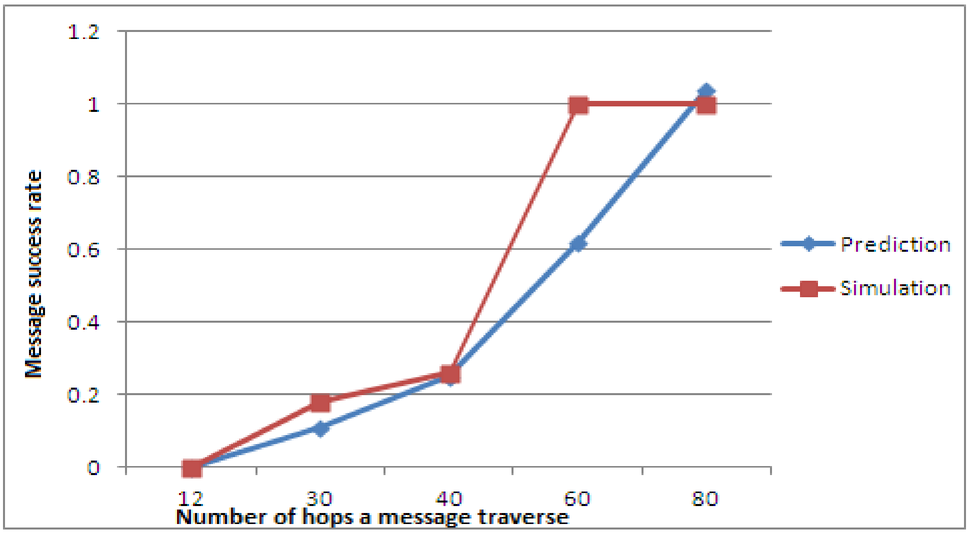


Figure 6. 3 Message success ratio when a 7*7 grid WSN employs BRALB for routing

For Figure 6.4 the message success rate is taken when the source and the sink are 18 hops apart. From the graph it is clearly seen that there is a consistently close correlation between the predictions and the simulated results. One hundred percent agreement between simulation results and predictions is obtained at larger hop counts.

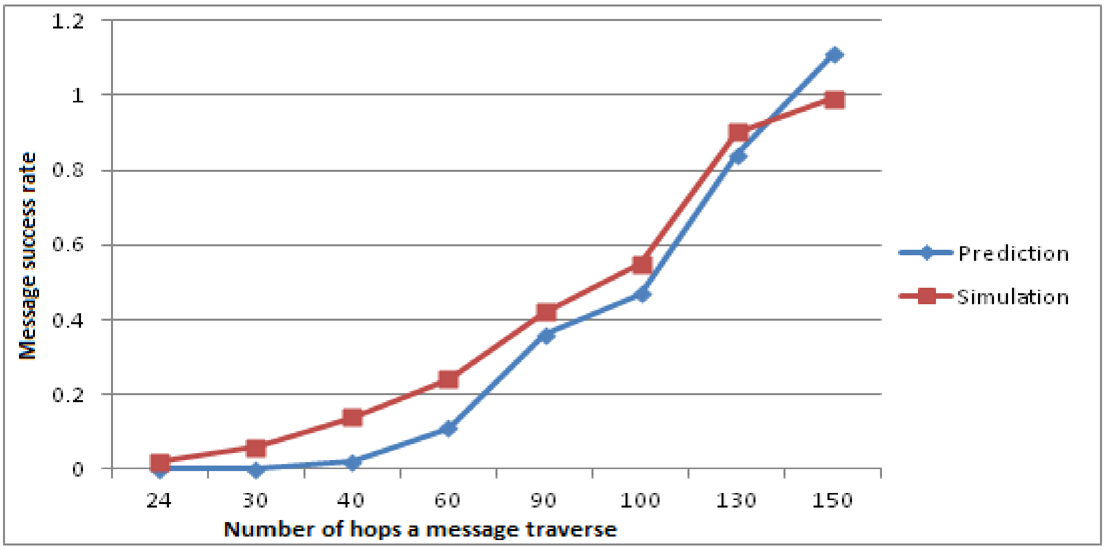


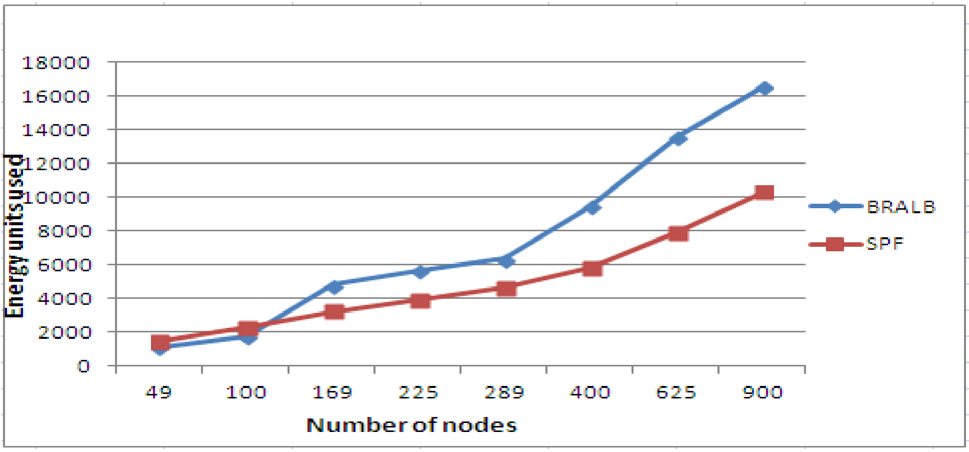
Figure 6. 4 Message success ratio when a WSN of 100 nodes (10*10 grid) employs BRALB

The simulation results clearly show that BRALB is a feasible algorithm. However, by closely observing the simulated results, it is very apparent that *as the grid size increases, the number of hops for a successful transmission also increases making BRALB unscalable*. This problem is addressed by adding a clustering mechanism to BRALB thereby reducing the delay and thus making the algorithm scalable.

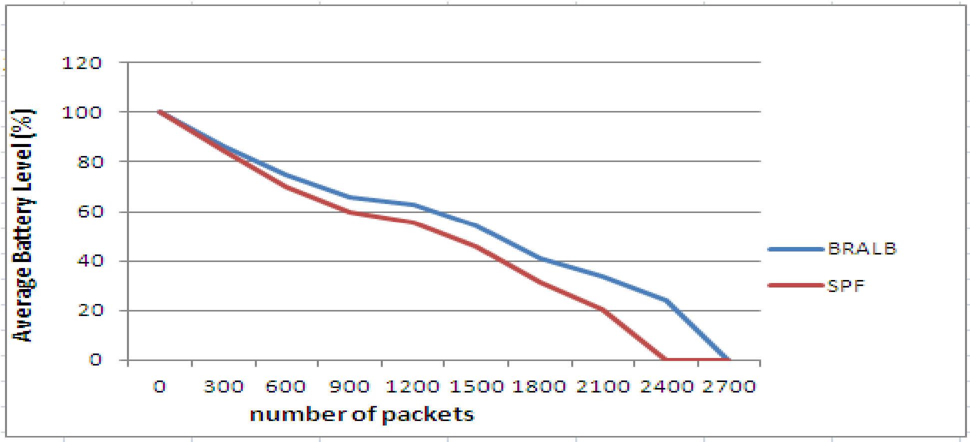
6.2.3.2 BRALB Versus Shortest Path First Algorithm in an m*m Square Grid

A. Energy Consumption

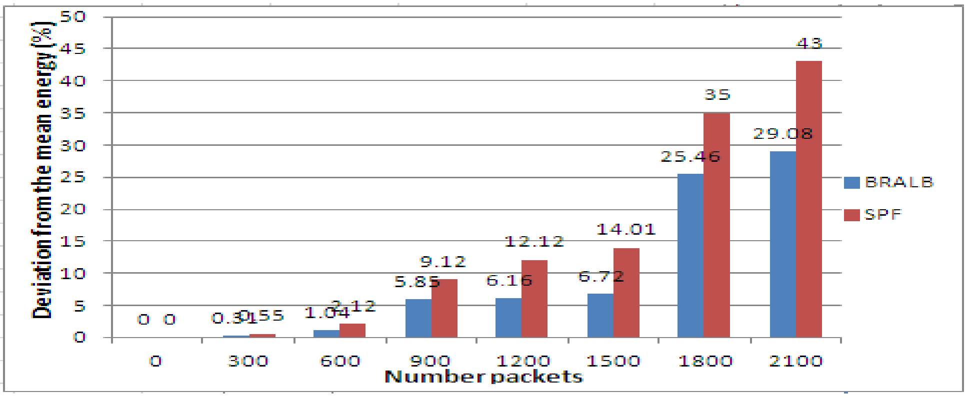
The energy consumption is defined as the total amount of energy used in the network. The total amount of energy is equivalent to the total number of hops the messages traversed. Through the simulation results it is demonstrated that BRALB uses roughly the same amount of energy as SPF while outperforming SPF in its ability to load balance the traffic across nodes as shown in Figures 6.5 (a) to (c).



(a)



(b)



(c)

Figure 6. 5 (a) Total energy used (b) average remaining battery level of nodes (c) node energy deviation from the mean.

Figure 6.5 (c) shows the result for the percentage deviation of the nodes' energy above the nodes' mean energy. The percentage deviation shows how far a measurement, on average will deviate from the average (mean). From the results it can be clearly seen that remaining energy for most of the nodes' is very close to the mean. This low percentage deviation demonstrates that the proposed routing algorithm is adept at balancing the traffic load across the network.

B. Delay

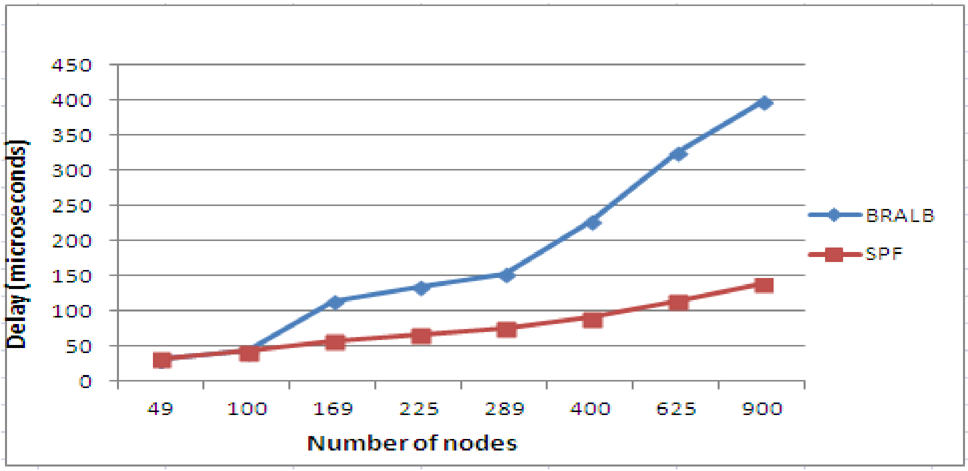


Figure 6. 6 Average packet delay of BRALB vs SPF

The delay experienced by the data packet as shown in Figure 6.6 is measured as the packet transmission time which is equal to packet size divided by bit rate. The packet size in BRALB is 600 bits and the bit rate of 250 kbps of Mica is assumed which gives a delay of 2.4 msec per hop.

The delay refers to the time it takes for a message to be routed from the source node to the destination node. The most significant delay in most networks including WSN is the time the devices take before sending the message to the medium of transmission but not the time delay in traversing the medium; as it travels at the speed of light. For this reason the *delay is expressed as the number of hops in each route rather than the actual time it takes to traverse the route*.

The average delay in BRALB is 250 microsecs worse than SPF for a network size of 900 nodes. As the network size increases SPF protocol tends to outperform BRALB in terms of average delay; this is expected, as SPF algorithm routes packets to the destination using the least number of hops. For BRALB this is not an issue as it is targeted for applications such as environment monitoring which are not very sensitive to delays.

C. Data Packet Loss Rate

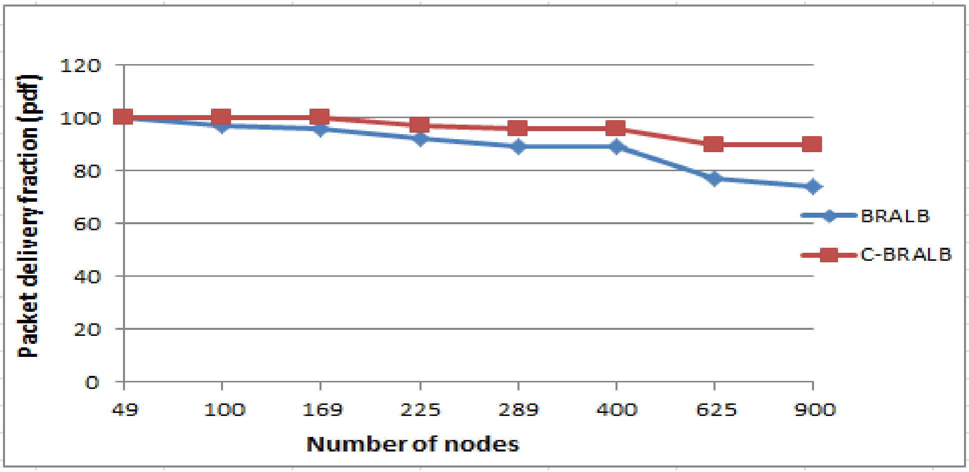


Figure 6. 7 Packet delivery ratio of BRALB vs SPF

The data loss rate is expressed in terms of the ratio of the total number of messages or packets that reach the destination node to the total number of messages or packets that were sent from the source node. There are less packet losses for the SPF as it has a complete knowledge of the whole network and so can always find the destination as long as there are no dead nodes.

The pdf (packet delivery ratio) is the ratio of messages received to messages sent in the network. As shown in Figure 6.7, BRALB has more packet loss as packets are not allowed to route infinitely. Each packet is calibrated to be discarded once it has consumed more energy than a normal SPF packet would have. This is called the hop threshold for a packet to live similar to Time-To-Live (TTL) in certain algorithms. When the number of nodes is less than 100, BRALB tends to outperform SPF due to the fact that with a small network diameter, the chances of routing a packet successfully to the sink by means of random walk increase. Though, as the network increases in size SPF tends to be more energy efficient, however, the packets lost rate here is not significant considering the gain in using limited computational power over SPF where knowledge of the whole network is required.

D. Fault Tolerance

Finally, in Figure 6.8 the robustness of BRALB was tested by randomly failing nodes and noting whether it was able to cope with the fault and deliver more than 70% of the traffic. This experiment was carried out for network sizes ranging from 49 to 361. This clearly shows that BRALB is very resilient and is able to function even when under random attacks.

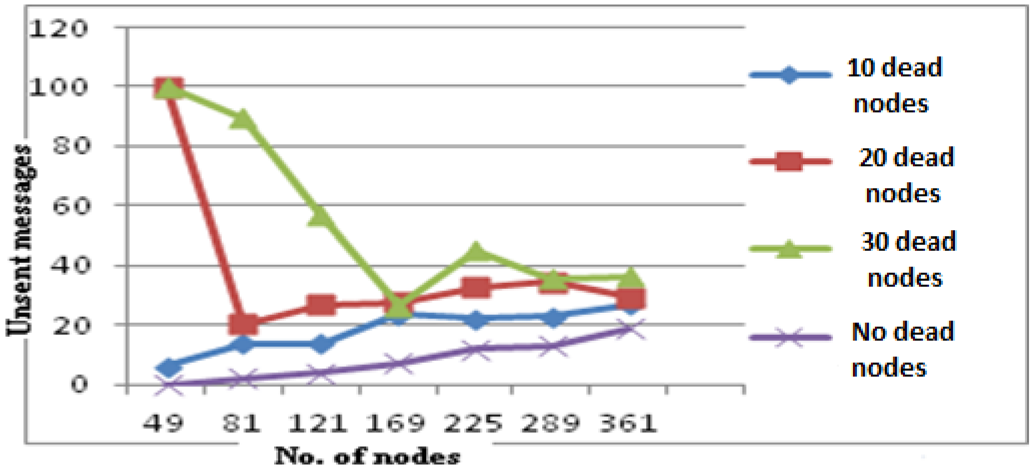


Figure 6. 8 Fault Tolerance property for BRALB.

E. Number of Dead Nodes

The study of the number of nodes alive and the nodes' remaining energy levels in a network running any algorithm can easily show the effectiveness of the algorithm. The goal of any effective routing protocol is to avoid over-utilising a set of nodes which may result in network partitioning. One effective way of avoiding this is to load balance the traffic among the nodes in the network. One way of checking how well the routing protocol load balances traffic is by analysing Figures 6.8 and 6.9 which clearly show that BRALB is effectively balancing the load across the network. In Figure 6.8, it is seen that even with random failing of nodes; BRALB can easily converge and almost maintain the same pdf, as if now nodes were dead. This is due to the redundancy created by BRALB as a result of its load balancing scheme.

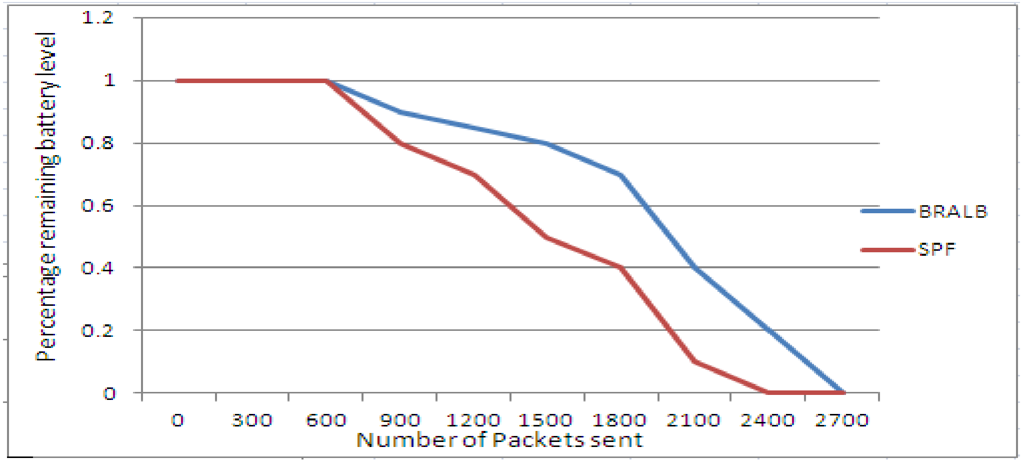


Figure 6. 9 Remaining number of live nodes against the total number of messages sent in the network.

6.3 Simulation of the Proposed C-BRALB Model

BRALB routing algorithm is not very suitable for large scale deployment due to higher energy requirements and increased delays. So, a cluster based version of BRALB, namely C-BRALB has been proposed to address this issue and improve the performance. In C-BRALB a node can exist in the following states, namely: ordinary nodes (ORN), cluster head node (CHN), cluster member node (CMN), candidate cluster head node (CCHN), and finally border node (BN). All nodes start as ordinary nodes and they may transition into other states after the cluster formation protocol process. A CMN is a node that has joined a cluster while a BN node is a node that has joined more than one cluster. A CCHN is a node that has declared itself as a candidate cluster head node and will transition to a CHN when other nodes accept its candidacy. In C-BRALB when a node has a data to transmit it sends it to its cluster head. It is the duty of the cluster head to aggregate traffic and to transmit it to the base station via the neighbouring cluster heads. Two Cluster Head nodes (CHNs) are neighbours when they can communicate with each other. There is always one CHN in a cluster at a time. A CHN keeps a counter for each neighbour CHN where it records the number of messages it had sent or received from them. When a CHN wants to forward data, it inspects its neighbour CHN counters and then forwards the data *to the CHN whose counter has the least number of messages*. The clustering mechanism in C-BRALB is used in this case as a means of shortening the network depth and for handling the peculiarities of a large area sensor network.

The CHNs (Cluster Head Nodes) forward the data gathered from the sensor nodes in their respective clusters as well as data gathered from other neighbouring CHNs towards the destination. Hence, the CHNs form a kind of a higher level backbone network among themselves and use multi-hop paths for routing data to the sink.

In C-BRALB, the number of clusters formed is based on the number of nodes in the network. In this protocol, once clusters have been formed, the CHNs form a multi-hop routing backbone based on the stipulated forwarding mechanism of C-BRALB. In multi-hop routing, the data packets received from the source node are transmitted to the intermediate nodes until they reach the destination in order to reduce the transmission energy consumption.

For intra-cluster data communication, every member node forwards the data to the CHN directly, whereas for sending the data towards destination, inter-cluster hierarchical routing is adopted to decrease the latency and load balance the traffic. The data travels from one cluster to a neighbour cluster with a CHN having the highest energy. This reduces latency. So, this approach is much better than traditional multi-hop routing between nodes.

Furthermore, in traditional multi-hop model, every intermediate node performs data aggregation resulting in huge delays whereas in cluster based models, only cluster heads perform data aggregation. Therefore, cluster based model is more suitable for time critical applications than the traditional multi-hop models. The proposed architecture was evaluated and its performance compared with that of LEACH the well-established cluster based algorithm.

6.3.1 Simulation Conditions for C-BRALB

All the assumptions made for the simulation of BRALB protocol also apply to C-BRALB. The following additional assumptions are also made:

(1) The nodes are randomly deployed.

(2) The cost for both Intra and inter cluster communications is equal.

(3) Nodes within a cluster can reach the cluster head in one hop.

(4) A CHN can reach its neighbour CHN in one hop communication.

(5) Cluster heads are aware of their neighbouring cluster heads.

6.3.2 Simulation Setup for C-BRALB

The same radio model assumed for BRALB has been used for simulating C-BRALB. Along with all the simulation parameters used for BRALB the following additional parameters shown in Table 6.2 have also been used for C-BRALB. NetLogo simulator is used in order to evaluate and compare C-BRALB, the proposed routing algorithm, with BRALB. NetLogo is well suited for modelling complex systems developing over time. It has a built-in library with sample models for guidance. There is even a sample model for Networks showing the implementation of the directed diffusion algorithm and preferential attachment thus it is suitable for modelling routing protocols. The environment used to test the performance of these two algorithms was modelled using NetLogo's graphic design tool in order to simulate a network. By using NetLogo, network parameters were varied in order to study their effect on the overall performance of each algorithm and do the comparison. The simulator facilitates to deploy the number of resource-constrained nodes and their required topology connectivity. Each simulation point is generated by averaging results from 20 simulations.

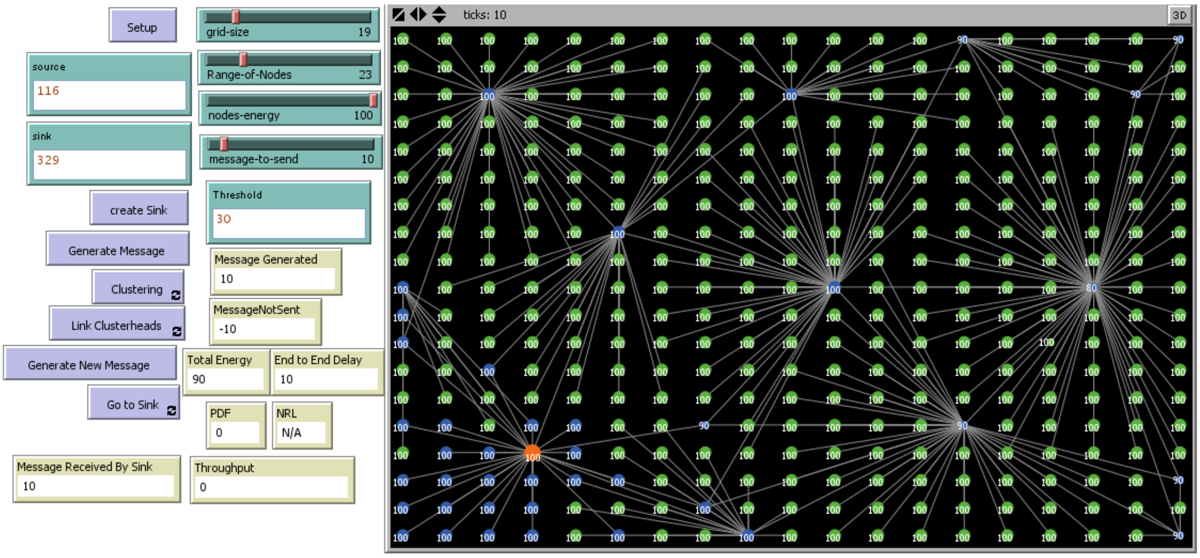


Figure 6. 10 Simulation snapshot for C-BRALB

Table 6. 2 C-BRALB Simulation Parameters.

C-BRALB Assumptions	
Threshold energy (TH_e)	5 KJ
Cluster Threshold energy (TH_c)	Half node energy before assuming cluster role
Cluster membership message (mem_msg)	600bits (75 bytes)
Minimum Level of Battery Required for Node to become cluster head	300 µJ

6.3.3 Simulation Results And Analysis for C-BRALB

C-BRALB is simulated and its performance has been measured on the basis of various network parameters such as pdf (packet delivery fraction), average energy dissipation, system lifetime, scalability and number of live nodes.

6.3.3.1 C-BRALB Versus Theoretical Prediction

In (Touray et al. 2011) equations 6.1 and 6.2 were developed. Equation 6.1 gives us the total probability of a successful transmission within a given number of hops using BRALB protocol.

$$p\{d \leq H\} = \sum_{i=0}^{\frac{H-k}{2}} \binom{k+2i}{i} \left(\frac{k+i}{k+2i} \cdot \frac{5}{9} + \frac{i}{k+2i} \cdot \frac{4}{9}\right)^{k+i} \cdot \left(\frac{k+i}{k+2i} \cdot \frac{4}{9} + \frac{i}{k+2i} \cdot \frac{5}{9}\right)^{i} \qquad (6.1)$$

$$\mathrm{d} = \mathrm{k} + 2.\mathrm{i} \qquad (6.2)$$

Where ‘d’ is the total number of hops traversed, k the shortest number of hops between the source and the destination and ‘i’ the total number of hops in the wrong direction. The percentage of successful transmissions could be obtained for various values of the number of hops traversed by the packet as shown below.

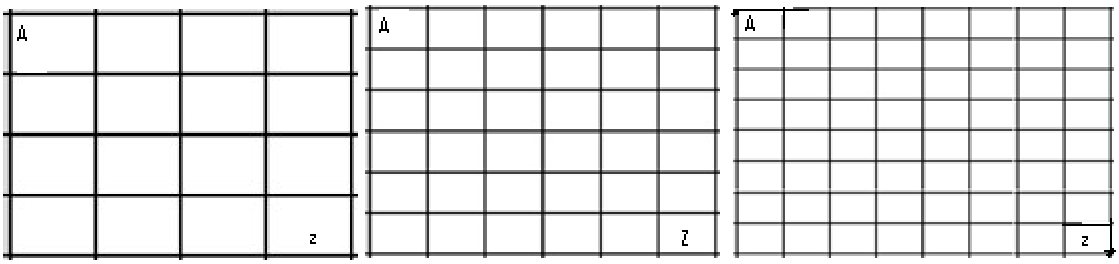
$$\mathrm{p}\{\mathrm{d} = 12\} = \binom{12}{3}\left(\frac{9}{12}.\frac{5}{9} + \frac{3}{12}.\frac{4}{9}\right)^9.\left(\frac{9}{12}.\frac{4}{9} + \frac{3}{12}.\frac{5}{9}\right)^3$$

$$\mathrm{p}\{\mathrm{d} \leq 6\} = 0.02940119411$$

$$p\{d \leq 24\} = 0.7401902058$$

$$p\{d \leq 30\} = 1.014642274$$

Let us derive the formula of obtaining the number of hops between *farthest* nodes in a square grid taking the shortest path.



(a) (b) (c)

Figure 6. 11 (a) 5*5 square grid (b) 7*7 square grid (c) 9*9 square grid

Let K_{max} be the maximum number of hops between two nodes on a square grid or a regular topology. If for each vertex there is a node then the maximum number of hops from node A to node Z for topologies shown in Figure 6.11 is calculated as follows:

In Figure 6.11 (a), K_{max}= 2*5-2 =8

In Figure 6.11 (b), K_{max}= 2*7-2 =12

In Figure 6.11 (c), K_{max}= 2*9-2 =14

If the total number of nodes is n then the formula for obtaining the maximum number of hops K_{max} in BRALB which uses a square grid is:

$$K_{max} = 2 * \sqrt{n} - 2 \qquad (6.3)$$

For C-BRALB using a regular network topology with x nodes in a cluster where the total number of nodes in the network is n, then the total number of clusters is $\frac{n}{x}$. The quotient of $\frac{n}{x}$ is the total number of clusters which gets its maximum value when all the clusters are in a straight line. This is a very unlikely scenario in a randomly deployed network such as the sensor network. Therefore K_{max} is most likely to be smaller than $\frac{n}{x}$.

The total number of clusters is the maximum possible shortest hop that can exist between the two farthest clusters, in any random topology. This unlikely scenario happens only when all the clusters are in a straight line.

Therefore in C-BRALB the maximum number of hops is given by:

$$K_{max} = \frac{n}{x} \qquad (6.4)$$

Where x is the number of nodes in a cluster and n the total number of nodes in the network.

Equation 6.4 gives the maximum possible value for K_{max} in any random deployment and can be used as a bench mark in order to avoid the network depth of C-BRALB being greater than BRALB.

Let us substitute the value of x as 10 in equation 6.4 and compare it with equation 6.3 in order to compare the network depths between that of BRALB and of C-BRALB for a given number of nodes.

$$0.1n \equiv 2 * \sqrt{n} - 2 \qquad (6.5)$$

Where the RHS of the above equation represents the maximum number of hops (network depth) for BRALB and the LHS represents the network depth for C-BRALB.

So, for n = 100 in equation 6.5 we have

$$10 \equiv 18 \tag{6.6}$$

Equation 6.6 means a network depth of 10 for C-BRALB compared to a network depth of 18 for BRALB.

And for n = 400 in equation 6.5 we have

$$40 \equiv 38 \tag{6.7}$$

Equation 6.7 means that a network employing C-BRALB will have a network depth of 40 compared to 38 when employing BRALB.

This gives an idea of how much the cluster size needs to be for a given number of nodes so as to avoid having a network depth greater than that of BRALB. For n=100 it is clear that a cluster size of 10 nodes in C-BRALB will always produce a network depth of 10 or less, which is less than the network depth (18) of BRALB. However, with 400 nodes, it is clearly seen that the optimum cluster size for C-BRALB needs to be more than 10.

The maximum number of nodes in a WSN for a cluster size of 10 for C-BRALB and BRALB to get the same network depth K_{max} can be obtained by manipulating equation 6.5 as shown below:

$$\frac{n}{10} = 2 * \sqrt{n} - 2$$

$$0.1n = 2 * \sqrt{n} - 2$$

$$0.1n + 4 = 2 * \sqrt{n}$$ Squaring both sides

$$0.01n^2 + 0.8n + 16 = 4n$$

$$0.01n^2 + 0.8n - 4n + 16 = 0$$

$$n^2 + 80n - 400n + 1600 = 0$$

$$n^2 - 320n + 1600 = 0 \tag{6.7}$$

Solving the above quadratic equation 6.7, the value of n could be either 314.9 or 5.1. Between these two values, n =315 is an acceptable value.

For x=20 and n=100, K_{max} for C-BRALB = 5 in equation 6.4 and K_{max} for BRALB=18 in equation 6.3.

To find the maximum number of nodes whereby the network depth (K_{max}) for C-BRALB will be less than or equal to that of BRALB; by equating equation 6.3 to equation 6.4 and solving the resulting equation by substituting x =20.

$$\frac{n}{20} = 2 * \sqrt{n} - 2$$

$$n = 40 * \sqrt{n} - 40$$

$$n + 40 = 40 * \sqrt{n}$$

$$n^2 + 80n + 1600 = 1600n$$

$$n^2 - 1520n + 1600 = 0$$

$$n \approx 1518 \tag{6.8}$$

We can also get the minimum cluster size for C-BRALB to match the network depth of BRALB in any network by equating equations 6.3 and 6.4. For simplicity we can neglect 2 in equation 6.3 for large values of 'n'.

$$\frac{n}{x} = 2 * \sqrt{n} - \cancel{2}$$

$$x^2 = \left(\frac{n}{2 * \sqrt{n}}\right)^2 = \frac{n^2}{4n} = \frac{n}{4}$$

Therefore

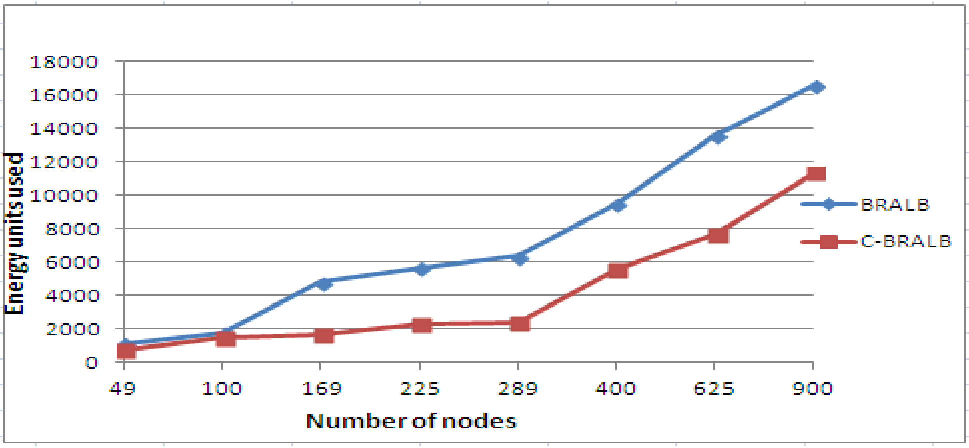
$$x = \frac{\sqrt{n}}{2} \tag{6.9}$$

Substituting n=100, in the above equation 6.9, x= $\frac{\sqrt{100}}{2}$ = 5

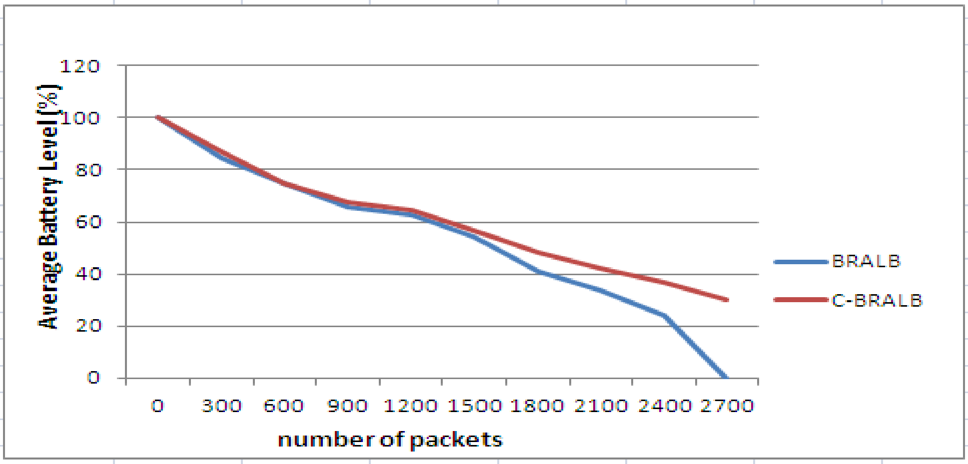
This gives a rough estimation of the ideal cluster size. The formula $x = \frac{\sqrt{n}}{2}$ can be used as a baseline for determining the appropriate cluster size to use for a given number of nodes.

A. Energy Consumption

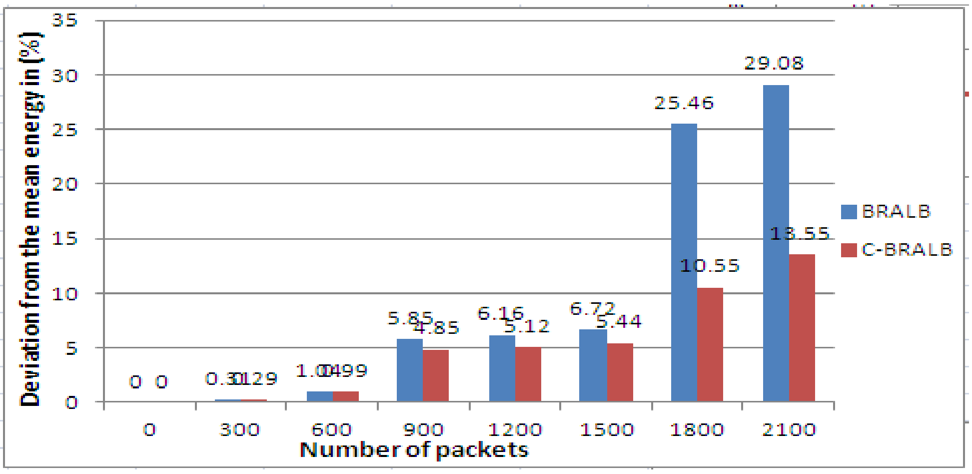
The energy consumption considered for evaluation here is the total amount of energy used in the network for routing purposes and maintenance. The total amount of energy is equivalent to the total number of hops the messages (both data and routing) have traversed, as each hop is assumed to use at least one unit of energy.



(a)



(b)



(c)

Figure 6. 12 (a) Total energy used (b) average remaining battery level of nodes (c) node energy deviation from the mean for C-BRALB vs BRALB.

Figure 6.12 (a) shows the total energy consumed by both BRALB and C-BRALB for transmitting 2100 packets of data over the simulation time. It is clearly evident that for network sizes of less than 100 nodes the energy utilisation of both algorithms is almost the same, however, the performance of C-BRALB in terms of energy utilisation becomes significantly better as the network size increases beyond 100 nodes. The reason for this is that for small networks the benefit of clustering is negated by the resulting overheads generated by the clustering process while for larger networks the gain in clustering far outweighs the resulting overheads. In Figure 6.12 (b) the network employing C-BRALB is again shown to save 10% more energy than that employing BRALB in terms of remaining battery life as less energy is consumed by the network while employing C-BRALB algorithm. In Figure 6.12 (c), the energy deviation from the mean in WSN employing C-BRALB is in the range of 0.3% to 13.6% while that for BRALB is in the range of 0.3% to 29.1%. This clearly shows that C-BRALB results in a better load distribution across the network than BRALB. The effectiveness of C-BRALB over BRALB gets more pronounced as more messages are generated. This is because the cost saving becomes more pronounced due toclustering in C-BRALB for large number of data packets. The significant improvement in the energy efficiency demonstrated by C-BRALB is again due to the clustering incorporated in C-BRALB.

B. Percentage Packet Delivery Ratio

Packet Delivery Ratio/Fraction (PDF) is the ratio between the number of packets delivered to the receiver and the number of packets sent by the source. In Figure 6.13 the difference in PDF performance between BRALB and C-BRALB gets more significant as the network size increases. The PDF for C-BRALB remains consistently above 90% for varying network size, while that of BRALB can be as low as 70% for large networks. This superior PDF performance of C-BRALB is due to the clustering mechanism, which decreases the network depth and hence increases the probability of the packets reaching the sink.

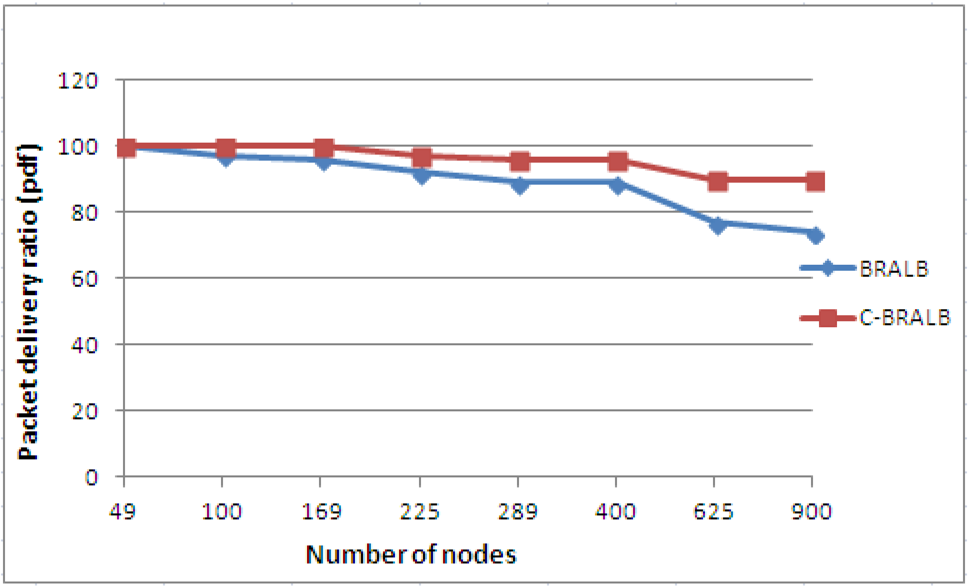


Figure 6. 13 Percentage packet delivery ratio for varying number of nodes

C. Packet Delivery Delay

The delay parameter used in Figure 6.14 is measured as the packet transmission time which is equal to the packet size divided by the bit rate. The packet size in C-BRALB is 600 bits and the bit rate of 250 kbps of Mica sensor node is assumed which gives a delay of 2.4 msec per hop.

The delay refers to the time it takes for a message to be routed from the source node to the destination node. The most significant delay in most networks including WSN is the time the devices take before sending the message to the medium of transmission but not the time delay in traversing the medium (as the message travels at the speed of light). For this reason the delay is expressed as the number hops in each route as opposed to the actual time it takes to traverse the route.

In Figure 6.14, in small networks, say with a size of less than 100 nodes the delay difference between BRALB and C-BRALB is less significant than in larger network sizes, this is due to the fact that the clustering mechanism is more effective in reducing the network depth as the network gets larger. In Figure 6.14 the delay for C-BRALB is less than 50% that of BRALB due to clustering, and hence allowing the network to scale without any drastic effect on the network.

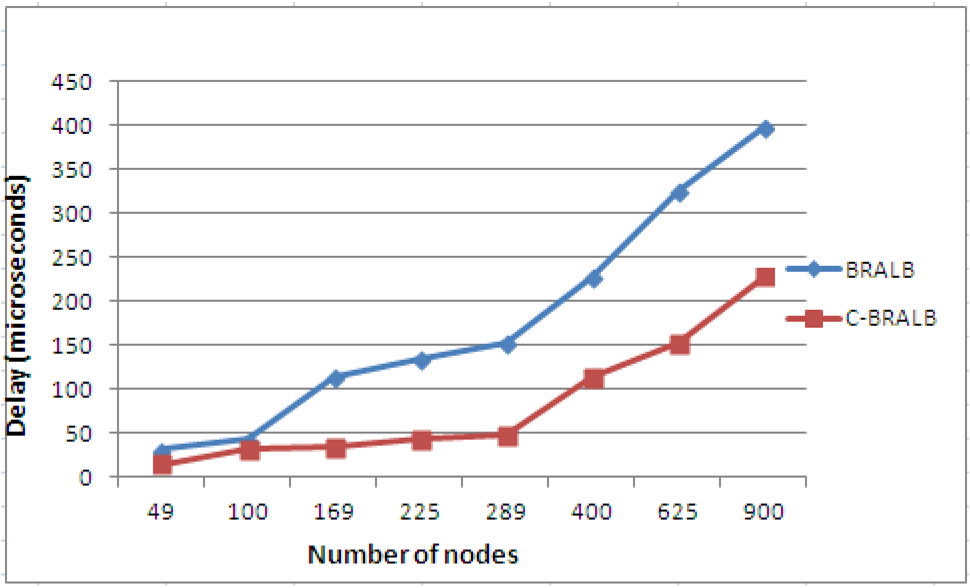
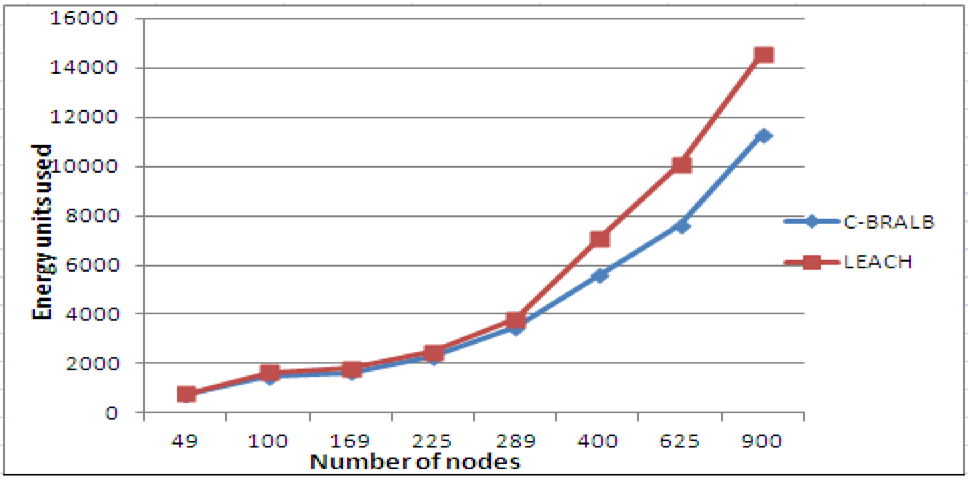


Figure 6. 14 Average packet delay for varying network sizes

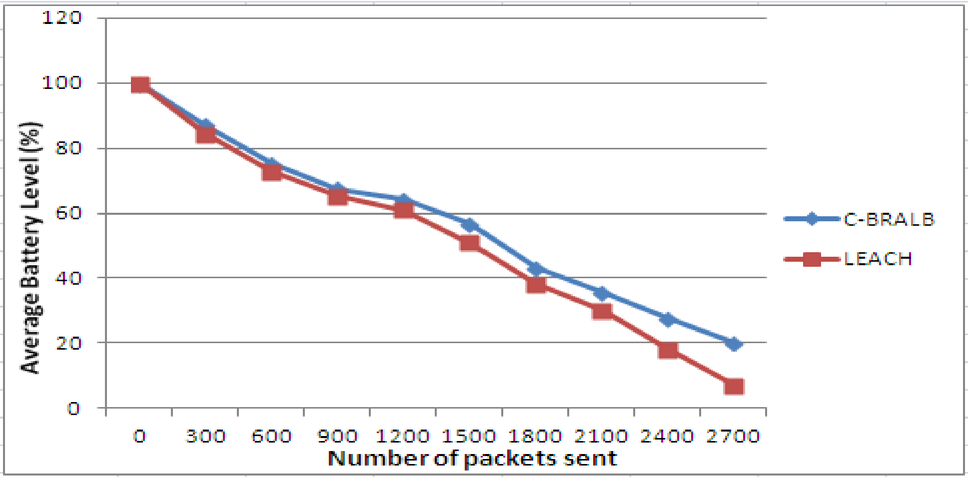
6.3.3.2 C-BRALB Versus LEACH Protocol

A. Energy Consumption of C-BRALB vs. LEACH

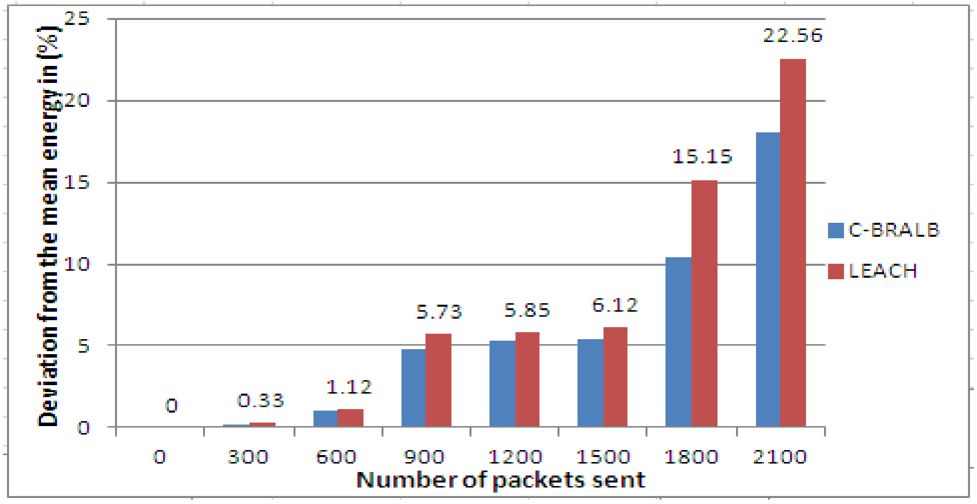
The energy gain of C-BRALB compared to LEACH is demonstrated in Figure 6.15. In Fig. 6.15 (a) it is shown that for networks having less than 289 nodes the energy conservation demonstrated by C-BRALB over LEACH is almost insignificant, however, as the network increases in size beyond 289 nodes there is a significant increase in energy consumption by LEACH protocol as it consumes 25% more energy than C-BRALB. This is mainly due to the fact that C-BRALB performs multi-hop data transmission to the sink while *LEACH does one hop transmission* to the sink which is less energy efficient. This is confirmed in both Figures 6.15 (b) and (c). This fact is also backed by the finding in the literature review that multi-hop transmission is more energy efficient than long range single hop transmission.



(a)



(b)



(c)

Figure 6. 15 (a) Total energy used (b) average remaining battery level of nodes (c) node energy deviation from the mean for C-BRALB vs. LEACH

Figure 6.15(c) gives the graph of the standard deviation (in percent) of the nodes' energy from their mean values. The standard deviation shows how far a measurement, on average will deviate from the mean value. From the results it can be clearly seen that most of the nodes' remaining energy are very close to the mean value when the sensor network uses C-BRALB with much less deviation than when using LEACH protocol. This clearly shows that C-BRALB is load balancing the traffic across the network better than LEACH.

B. Percentage Packet Delivery Ratio of C-BRALB vs. LEACH

In Figure 6.16 the PDF for both algorithms is more than 90% for network sizes of less than 400 nodes. For network sizes of up to 400 nodes the PDF of LEACH is better than that of C-BRALB by 2%. However, for network sizes greater than 400 nodes C-BRALB PDF is better than that of LEACH by roughly 5%. This is again due to the fact that LEACH employs one hop data transmission to the sink, therefore for large networks there is the tendency of packet being dropped as their distance from the sink is far beyond the transmission range of the nodes.

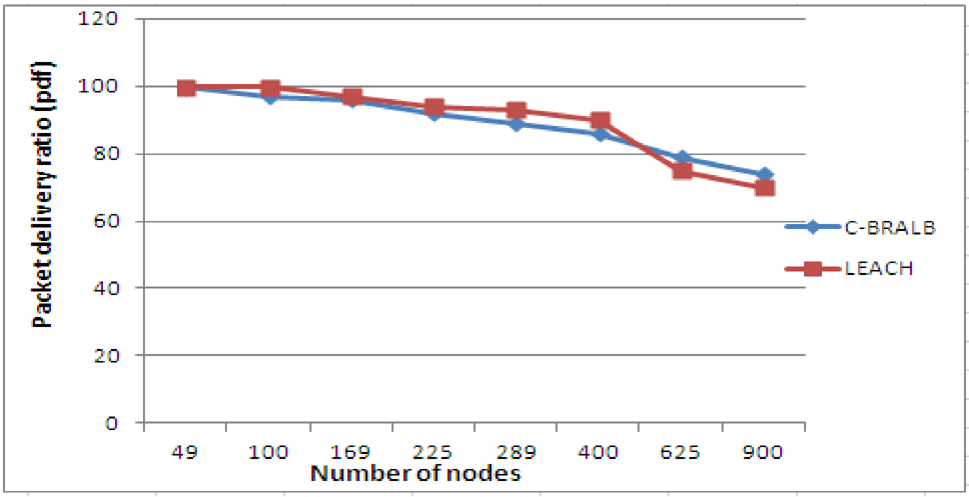


Figure 6. 16 Packet delivery ratio of C-BRALB vs. LEACH

C. Packet Delivery Delay of C-BRALB vs. LEACH

Figure 6.17 shows a constant delay of 25ms for LEACH while for C-BRALB the delay increases to a maximum of 240ms as the network size increases to 900 nodes. The delay in LEACH is very small due to the fact that packets are either dropped or transmitted by one hop to the sink. The delay for C-BRALB is higher than that of LEACH; however it does not affect the performance of the network as it is within the acceptable delay range.

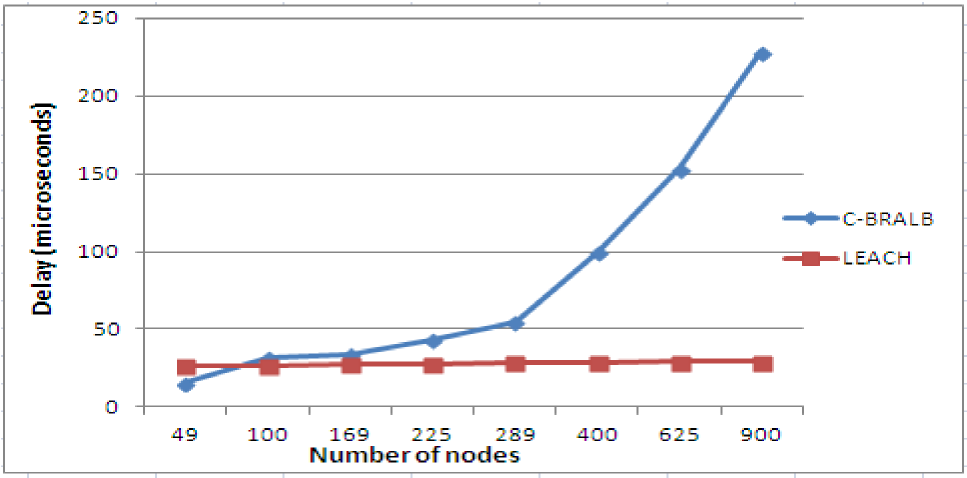
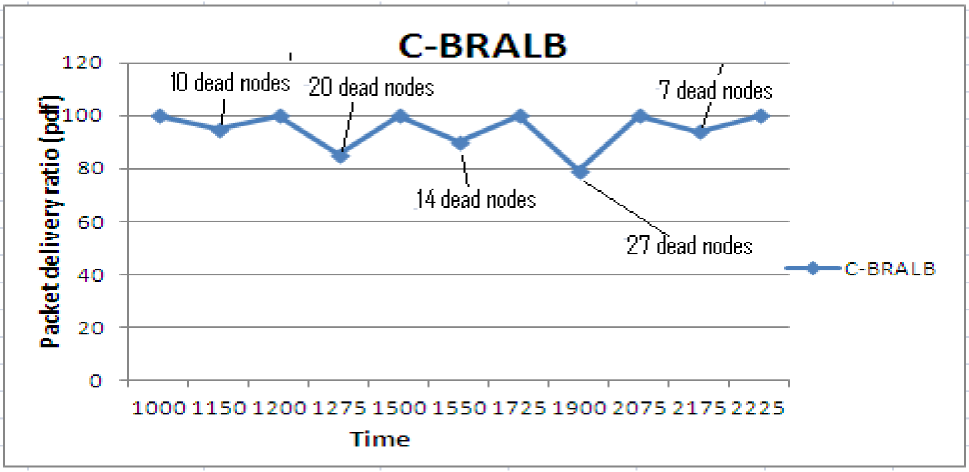


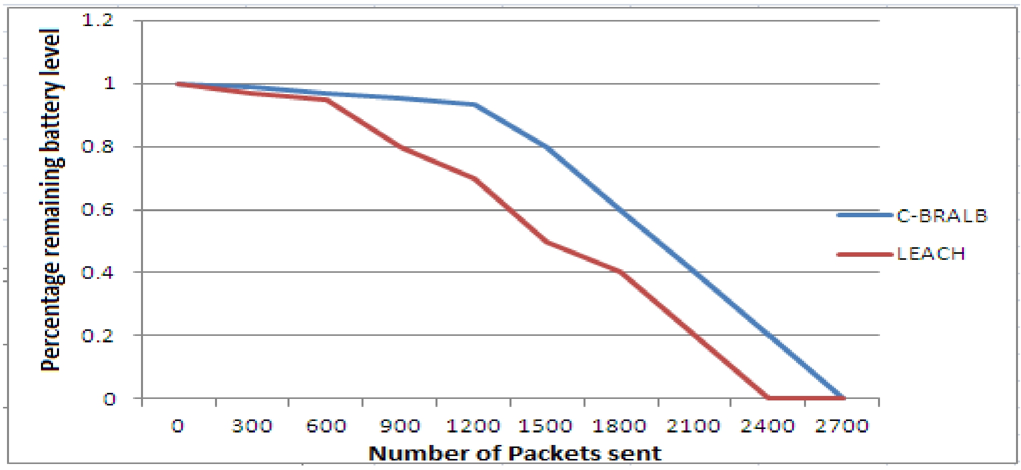
Figure 6. 17 Average packet delay of C-BRALB vs. LEACH

D. C-BRALB Fault Tolerance

To measure the resilience, reliability and robustness of C- BRALB routing mechanism, the simulation is extended to investigate how the pdf of the network will be affected by a random attack on the network. To do this, a random number of nodes were failed and the time taken for the network to converge has been measured.



(a)



(b)

Figure 6. 18 (a) Fault Tolerance property for C-BRALB (b) Remaining number of live nodes against the total number of messages sent in the network.

Node failure can occur due to many reasons which include attack at cluster level, electronic fault or energy depletion at individual node level. As seen from Figure 6.18 (a), the pdf decreased dramatically when nodes failed. However, the network starts to reorganise its routing paths over time and dynamically regains the same pdf value before the nodes failure. Thus, the proposed C-BRALB scheme is efficient, reliable, and resilient and it is robust against random errors or attack. Thus C-BRALB enables the network to dynamically and efficiently reorganise and heal it-self against node failure or attack. This is further justified by Figure 6.18 (b) in its ability for C-BRALB to conserve energy by 20% to 30% for increasing data traffic, compared to the LEACH protocol.

6.4 Summary

The proposed architecture has been compared with two existing routing algorithms SPF and LEACH. Generally, there are two evaluation scenarios to successfully obtain the true behaviour and performance of a new algorithm. In this chapter, the fixed network evaluation and scalability analysis are used to evaluate the performance of the proposed routing algorithm. The performance was measured by altering network parameters such as node density, data volume, dead nodes etc. The proposed architecture is more energy efficient, scalable and has a better pdf and extended network lifetime. The performance parameters pertaining to various components and aspects of the proposed architecture have been simulated and analysed. The analysis of energy consumption in various activities of clustering and clustering process has also been discussed in detail. The main emphasis in this chapter of the thesis has been on designing a new model that handles the problem of energy conservation, scalability and robustness in the network layer of WSNs. In this chapter, the evaluation and performance of the proposed routing algorithm has been discussed and compared with the theoretical prediction and to two well-known protocols namely: SPF and LEACH. The comparison with the theoretical prediction served as the scalability analysis while the comparison with SPF and LEACH serve as the fix network evaluation. The performance of the proposed routing protocol (BRALB) under various scenarios and configurations has been examined. The statistical results were presented which correlates with the simulated results that verified that the proposed routing protocol is viable. Then the proposed routing protocol was modified with the inclusion of a clustering mechanism. Extensive Simulation results show that the resulting algorithm (C-BRALB) was more effective in conserving energy than the previous one. Thus, the proposed routing algorithm is efficient, scalable, and reliable.

The good performance of the proposed routing algorithm is due to the fact that it is simple but yet efficient enough to avoid the extra overheads and memory requirement which may result from a complex algorithm. First BRALB was proposed and then modified and then integrated with a clustering scheme that resulted in C-BRALB. The inclusion of a clustering scheme for load distribution which uses local knowledge to distribute and balance the load in the network has greatly enhanced C-BRALB. This was demonstrated in this chapter through the enhanced performance of C-BRALB over BRALB and LEACH. In addition, the random forwarding technique is improved by biasing the forwarding of data toward neighbour nodes with highest remaining energy instead of choosing them uniformly at random. Hence, the nodes' selection criteria will be based on a predefined criterion rather than selecting any neighbour at random.

Though similar techniques exist for routing mechanism, the proposed scheme *has the advantage of using a simple routing scheme based on random walk and local knowledge* to efficiently distribute the load. Moreover, this routing mechanism *does not require any global knowledge* apart from the initial flooding to create the clusters since its routing decision is based on random walk and local knowledge.

Chapter 7

CONCLUSION AND SUGGESTIONS FOR FUTURE WORK

Chapter 7

CONCLUSION AND SUGGESTIONS FOR FUTURE WORK

7.1 Discussion and Conclusions

A wireless sensor network (WSN) is one of the emerging technologies that are made possible by the recent developments in Electronics and Computer Science. A WSN is used in various applications as mentioned in this thesis to monitor a phenomenon of interest and then report the collected data to a base station. One of the main problems of WSNs is that the nodes are powered by small AA batteries which are most of the time not rechargeable and even when they do, it may not be feasible to recharge them due to their nature of deployment. Therefore, it is very crucial for the survival of the network to efficiently use the limited node energy. One way of achieving this is by developing an energy efficient routing algorithm to provide the mechanism and policies of data transmission within the WSN.

In this thesis, a biased-random algorithm namely BRALB and its clustering version (C-BRALB) are proposed to efficiently route data within a WSN. The proposed routing algorithms are made simple but yet efficient so as to avoid the extra overheads and memory requirement which may result from a more complex algorithm. First BRALB was proposed and modified and then integrated with a clustering scheme that resulted in C-BRALB. In this thesis, it is shown that BRALB algorithm relies on Random walk and local knowledge so as to forward data to the sink. However, from the results gathered it was shown that there was scalability issue with BRALB which was then successfully addressed by integrating a clustering scheme to BRALB. The developed routing algorithm *is based on energy biased random walk technique to forward data and to statistically estimate the remaining energy of neighbouring nodes*. Therefore, load balancing and path availability are achieved without the need to monitor the whole network.

In this research thesis, the working mechanism and principles of the proposed architecture are specified in great detail. The architecture is designed to work in different environments where the data to be sent is in comparison to the size of a normal hello packets of traditional routing protocols. In the literature, various clustering techniques have been proposed based on both the network size and node density. For small networks, a one hop communication and centralised clustering scheme is proposed to efficiently route packet to the sink. For larger networks hierarchical clustering schemes is proposed instead. In this thesis, C-BRALB uses the hierarchical clustering scheme for large networks and reverts to BRALB within small networks.

C-BRALB conserves the energy usage of the nodes by avoiding duplicate data to be sent through the network by the use of cluster head nodes which act as the central collection point for a cluster. The cluster head rotation of C-BRALB is based on maximum remaining energy of a node and therefore it avoids loss of data due to low energy nodes becoming cluster heads. It also achieves load balancing without the need for global network knowledge which results in high overhead and high energy usage and therefore leads to network partitioning. Moreover, the performance of C-BRALB has been demonstrated to become scalable and stable as the network becomes larger.

The proposed models have been simulated and compared to the statistical analysis and further compared to well-known similar algorithms (SPF and LEACH). In the literature it was demonstrated that the best method for evaluating an algorithm is to conduct both a fixed network evaluation and a scalability analysis. The comparison of the proposed algorithms to their statistical analyses serve as the scalability analysis while their comparison to other similar established algorithms serve as the fixed network evaluation. The results from the statistical analyses closely correlate to the simulation results. Simulation have been carried to analyse the proposed models and compare them to similar algorithms on the basis of factors like average energy usage, network system lifetime, packet deliver ratio (pdf), average delay, scalability and fault tolerance. The performance was judged by varying different sensor network features such as number of nodes, number of cluster heads, number of nodes within a cluster, number of randomly failed nodes etc.

In the evaluation of BRALB, the shortest path first algorithm has been used as the benchmark for measuring the performance while for C-BRALB, LEACH was used as the yard stick. As evident by the results, the proposed algorithms are both suitable for a wireless network deployment. For networks of less than 100 nodes BRALB is more energy efficient than SPF by 2% while for networks of more than 100 nodes SPF is more energy efficient than BRALB by 5%. The standard deviation for the nodes' energy in BRALB is within the range of 0.3% to 29.1% while for SPF, it is within 1.5% to 43%. BRALD was able to recover from unexpected node failures or random attacks of up to 30 dead nodes in a network of 361 nodes. It recovered from such attacks and in the worst case scenario, was able to deliver at least 70% of the messages. The average delay in BRALB is 250 microsecs, worse than SPF for a network size of 900 nodes. As the network size increases SPF protocol tends to outperform BRALB in terms of average delay; this is expected, as SPF algorithm routes packets to the destination using the least number of hops. BRALB algorithm was then developed further to alleviate the scalability issue resulting in C-BRALB algorithm. C-BRALB was first compared to BRALB and it was 10% more energy efficient. The standard deviation of the nodes' energy in C-BRALB is in the range of 0.3% to 13.6% while in BRALB, it is in the range of 0.3% to 29.1%. The delay of C-BRALB was reduced to less than 50% the delay of BRALB. C-BRALB was then compared to LEACH. C-BRALB was found to consume 25% less energy than LEACH. The PDF of both C-BRALB and LEACH is more than 90% for network of up to 400 nodes. The PDF of LEACH is better than that of C-BRALB by 2% for network of less than 400 nodes while for networks of more than 400 nodes the PDF of C-BRALB is better by 5%.

To conclude, the proposed BRALB and C-BRALB are novel techniques for routing in a wireless sensor network deployed in environmental monitoring scenario where the data packets are small in size. The proposed algorithms are very scalable and fault tolerant. Thus, the proposed algorithms generate a routing mechanism for wireless sensor networks which are scalable, self-organised, robust, and depend only on local information for packet forwarding.

The research work described in this thesis has resulted in Journals and international Conferences publications, which are included in the list of references (Touray et al. 2011; Touray et al. 2012c; Touray and Johnson 2012a; Touray and Johnson 2012b; Touray et al. 2013).

7.2 Future Work

The problem of global warming on our planet cannot be over emphasised as it has direct impact on our lives and environment. Recently, there have been reports of the glacier in the North Pole melting at a fast rate, floods everywhere and the snow mountain caps melting which might be the result of global warming.

The proposed routing architecture will be extremely useful in the designing of practical applications especially in monitoring the temperature reading in the North Pole or any mountain of concern. The proposed models need further tests in larger environment before being deployed for practical applications in the real world. There are some research challenges that need to be addressed before their practical application in the real world. For future work the following suggestions need to be considered and addressed.

(1) The proposed C-BRALB algorithm can be extended to incorporate *multi-hop intra cluster communication* which may improve the scalability of the algorithm and its efficiency.

(2) This may serve as a model or a starting point for developing more energy efficient routing algorithms in the future.

(3) The proposed algorithm can be modified and implemented for WSNs incorporating mobile nodes.

References

Agrawal, D. P., Lu, M., Keener, T. C., Dong, M. and Kumar, V. (August 2004). "exploiting the use of wireless sensor networks for environmental monitoring." *journal of the environmental management*. 35-41.

Ahmed, N., Kanhere, S.S. and Sanjay, J. (2005). "Probabilistic coverage in wireless sensor networks." P*roceedings of the IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)*. 672-681.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002). "A survey on sensor networks." *Communications Magazine, IEEE.* **40**(8): 102-114.

Baras, J. S. and Mehta, H. (March 2003). "A Probabilistic Emergent Routing Algorithm for Mobile Ad Hoc Networks." *Modeling and Optimization inWireless, Mobile, and Ad Hoc Networks (WiOpt 03).*

Beraldi, R., Baldoni, R. and Parakash, R. (2009). "Lukewarm Potato Forwarding: A Biased Random Walk Routing Protocol for Wireless Sensor Networks." *6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON '09, 22-26 June 2009.* 1-9.

Biswas, R., Jain, N., Nandiraju, N. and Agrawal D.P. (2005). "Communication architecture for processing spatio-temporal continuous queries in sensor networks." *Annales Des Télécommunications*. **60**(7-8): 901-927.

Boukerche, A. (2008). "*Algorithms and protocols for wireless sensor networks.*" Oxford : Wiley-Blackwell.

Braginsky, D. and Estrin, D. (2002). "Rumor routing algorthim for sensor networks." *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications.* Atlanta, Georgia, USA: ACM**.** 22-31.

Bulusu, N., Heidemann, J., and Estrin, D. (2000). "GPS-less low-cost outdoor localization for very small devices." *Personal Communications, IEEE.* **7**(5): 28-34.

Callaway, E. H. (2003). "*Wireless Sensor Networks: Architectures and Protocols (Internet and Communications).*" Boca Raton Florida:Auerbach Publications.

Capkun, S., Mounir, H., Hubaux, J. P. (2001). "GPS-free positioning in mobile ad-hoc networks." *Proceedings of the 34th International Conference on System Sciences,. Annual Hawaii,3-6 Jan. 2001, Hawaii.* 3-6

Chen, W.-P., Hou, J and Sha, L. (2004). "Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks." *IEEE Transactions on Mobile Computing.* **3**(3): 258-271.

Chim, T. W. (2005). "Along & across algorithm for routing events and queries in wireless sensor networks." *Proceedings of 2005 International Symposium on Intelligent Signal Processing and Communication Systems, ISPACS, 13-16 Dec. 2005*. 725-728.

Christian, B. (2002). "on the minimum node degree and connectivity of a wireless multihop network." *mobihoc 2002, ACM Press*. 80-90.

Cordeiro, C. D. M. and Agrawal, D. P. (2006). "*Ad hoc & sensor networks : theory and applications.*" Hackensack, NJ: World Scientific Publishing Co.

Cross bow Te c h n o l o g y , Inc. . (2008). "MICA2 data sheet". Available: http://www.xbow.com. Last accessed 17th April 2013.

Culler, D., Estrin, D., Srivastava, M. (Aug. 2004). "Guest Editors' Introduction: Overview of Sensor Networks," *Computer*. **37**(8):41-49.

Dam, T. V. and Langendoen, K. (2003). "An adaptive energy-efficient MAC protocol for wireless sensor networks." *Proceedings of the 1st international conference on Embedded networked sensor systems*. Los Angeles, California, USA: ACM**,** 171-180.

De, V. and Borkar, S. (1999). "Technology and design challenges for low power and high performance [microprocessors]." *1999 International Symposium Proc. on Low Power Electronics and Design, Aug. 1999*. 163-168.

Di, T. and Georganas, D. (2002). "A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks." *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. ACM: 32-41.

Estrin, D., Govindan, R and Heidermann, J. (1999). "Next Century Challenges: Scalable Coordination in Sensor Networks." *ACM Mobicom*. 263-270.

Förster, A., Förster, A. and Murphy, A. L. (2010). "Optimal Cluster Sizes for Wireless Sensor Networks: An Experimental Analysis". *Ad Hoc Networks 2009*. 49-63.

Gajbhiye, P. and Mahajan, A. (2008). "A Survey of Architecture and Node Deployment in Wireless Sensor Network." *In proceedings of First International Conference on the Application of Digital Information and Web Technologies, Ostrava, Czech Republic, August 4-6, 2008*.

Ghane, M. and Rajabzadeh, A. (2010). "Remaining-energy based routing protocol for wireless sensor network." I*nternational Symposium on Computer Architecture and Digital Systems (CADS), 2010 15th CSI*. 63-73.

Gruteser, M., Jain, A., Deng, J., Zhao, F. and Grunwald, D. (December 2001). "Exploiting Physical Layer Power Control Mechanisms in IEEE 802.11b Network Interfaces." *DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF COLORADO AT BOULDER.*

Guan, X. Guan, L., Wang, X.G. and Ohtsuki, T. (2009). "A Novel Routing Algorithm Based on Ant Colony System for Wireless Sensor Networks." *Proceedings of 18th Internatonal Conference on Computer Communications and Networks, ICCCN, 3-6 Aug. 2009*. 1-5.

Guo, W., Zhang, W. and Lu, G. (2010). "PEGASIS Protocol in Wireless Sensor Network Based on an Improved Ant Colony Algorithm." *Second International Workshop on Education Technology and Computer Science (ETCS), 6-7 March 2010* . 64-67.

Hahn, R. and Reichl, H (1999). "Batteries and power supplies for wearable and ubiquitous computing." *The Third International Symposium on Wearable Computers*. 168-169.

He, L. (2010). "Efficient Multi-Path Routing in Wireless Sensor Networks." *6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), 23-25 Sept. 2010*. 1-4.

Heile, B., Gifford, I. and Siep, T. (July, 1999). "IEEE 802 Perspectives, The IEEE P802.15 Working Group for Wireless Personal Area Networks." *IEEE Network*.

Heinzelman, W. (2000a). "*Application-Specific Protocol Architectures for Wireless Networks*." Ph.D. thesis, Massachusetts Institute of Technology.

Heinzelman, W. R., Chandrakasan, A. and Balakrishnan. (2000b). "Energy-efficient communication protocol for wireless microsensor networks." *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 4-7 Jan. 2000*.

Hill, J. Horton, M., Kling, R. and Krishnamurthy, L. (June 2004). "the platform enabling wireless sensor networks," communications of the ACM." *communications of the ACM*. **47**(6): 41-46.

Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D. and Pister, K. (2000). "System architecture directions for networked sensors." *SIGPLAN Not*. **35**(11): 93-104.

Howard, A., Mataric, M. J. and Sukhatme, G. S. (June, 2002). "Mobile Sensor Networks Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem." *International Symposium on Distributed Autonomous Robotics Systems, Japan.* 299-308.

Howard, A., Mataric, M. J. and Sukhatme, G. S. (September, 2002.). "An Incremental Self-Deployment Algorithm for Mobile Sensor Networks." *Autonomous Robots.*13(2):113-126.

Huang, C and Tseng, Y. (2003). "The Coverage Problem in a wireless Sensor Network," *in Proceedings of WSNA*. 115-121.

Ingalls, R. G. (2008). "Introduction to simulation." *In WSC '08: Proceedings of the 40th Conference on Winter Simulation (2008)*. 17-26.

Intanagonwiwat, C., Govindan, R. and Estrin, D. (2000). "Directed diffusion: a scalable and robust communication paradigm for sensor networks." *Proceedings of the 6th annual international conference on Mobile computing and networking*. Boston, Massachusetts, USA: ACM**.** 56-67.

Issariyakul, T. and Hossain, E. (2009). "*Introduction to Network Simulator NS2*." Milton Keynes, UK: Springer Science+Bussiness Media, LLC.

Jain, N. and Agrawal, D. P. (2005). "Current Trends in Wireless Sensor Network Design " *International Journal of Distributed Sensor Networks* . **1**(1): 101-122.

Jiang, J. and Dou W. (2004). "A coverage preserving density control algorithm for wireless sensor networks." *ADHOC-NOW'04*. 42-55.

Jiang, M., Li, J. and Tay, Y.C.. (1998). "Cluster Based Routing Protocol functional specification." Available: http://tools.ietf.org/html/draft-ietf-manet-cbrp-spec-01. Last accessed 25th April 2013.

John, M. Fulford-Jones, T.R.F., Bonato, P. and Welsh, M. (2005). "A wireless, low-power motion analysis sensor for stroke patient rehabilitation." *Biomedical Engineering Society (BMES) 2005 Annual Fall Meeting, Sept. 28–Oct. 1, Baltimore (MD)*.

Jovanov, E., Poon, C., Yang, G-Z., Zhang, Y.T. (Nov. 2009). "Guest Editorial Body Sensor Networks: From Theory to Emerging Applications," *IEEE Transactions on Information Technology in Biomedicine*. **13**(6):859-863.

Kahn, J. M., Katz, R. H. and Pister, K.S.J. (1999). "Next century challenges: mobile networking for “Smart Dust&rdquo." *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, Aug. 1999*. Seattle, Washington, USA, ACM**:** 271-278.

Kimel, J. C. (2005). "Thera-Network: a wearable computing network to motivate exercise in patients undergoing physical therapy." *25th IEEE International Conference on Distributed Computing Systems Workshops, 2005, 6-10 June*. 491- 495.

Konze, K. (1998). "Wildlife Monitoring Programs and Inventory Techniques for Ontario(TechnicalManualTM009)."Available:http://www.mnr.gov.on.ca/en/index.html?CSB_ic-name=topMenu&CSB_ic-info=home_Eng. Last accessed 25th April 2013.

Kumar, S., Lai, T. H. and Balogh, J. (2004). "On K-Coverage in a Mostly Sleeping Sensor Network." *Proceedings of the 10th annual international conference on Mobile computing and networking (MobiCom "04)*. New York, NY, USA: ACM, 144-158.

Kuorilehto, M., Hannikainen, M. and Hämäläinen, T.D. (2005). "A Survey of Application Distribution in Wireless Sensor Networks." *EURASIP Journal on Wireless Communications and Networking*. **2005**(5): 859712.

Luwei, J., Feng, L. and Li, Y. (2011). "Energy saving routing algorithm based on SPIN protocol in WSN." *International Conference on Image Analysis and Signal Processing (IASP), Oct. 2011*. 416-419.

Mainwaring, A. Culler, D., Polastre, J., Szewczyk, R. and Anderson, J. (Sept. 2002). "Wireless Sensor Networks for Habitat Monitoring." *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, WSNA '02, Atlanta, Georgia, USA*. New York, NY, USA:ACM, 88-97.

Meguerdichian, S. Koushanfar, F., Potkonjak, M. and Srivastava, M.B. (2001). "Coverage problems in wireless ad-hoc sensor networks." *INFOCOM Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*. **3:**1380-1387.

Min, R. K. (2003). "*Energy and Quality Scalable Wireless Communication*." Ph.D. thesis, Massachusetts Institute of Technology, USA.

Nagle, J. (1984). "RFC-896: Congestion Control in IP/TCP Internetworks." Available: http://www.ietf.org/rfc/rfc896.txt. Last accessed 22th April 2013.

National Semiconductor Corporation. (April 1999)." LMX3162 Evaluation Notes and Datasheet."Available:http://www.ti.com/general/docs/lit/getliterature.tsp?literatureNumber=snoa089&fileType=pdf. Last accessed 25th April 2013.

Nayak, A. and Stojmenovic, I (2010). "*Wireless Sensor and Actuator Networks:Algorithms and Protocols for Scalable Coordination and Data Communication*." Wiley-IEEE Press.

Pearlman, M. R. and Haas, Z. J. (1999). "Improving the performance of query-based routing protocols through “diversity injection." *Wireless Communications and Networking Conference, WCNC IEEE*. 1548-1552.

Powers, R. (1994). "Advances and trends in primary and small secondary batteries." *IEEE Aerospace and Electronics Systems Magazine*. **9**(4): 32-36.

Rachuri, K.K. and Murthy, C. S. R. (2011). "Energy efficient and low latency biased walk techniques for search in wireless sensor networks." *J. Parallel Distrib. Comput*. **71**(3): 512-522.

Roth, M. and Wicker, S. (2003). "Termite: ad-hoc networking with stigmergy." *Global Telecommunications Conference, GLOBECOM '03. IEEE, 1-5 Dec. 2003*. **5:** 2937-2941.

Roy, L. A. and Agrawal D. P., (Nov. 2003). "Wearable Networks: Present and Future," *IEEE Computer*. **36**(11):31-39.

Schmidt, R., Norgall, T., Mörsdorf, J., Bernhard, J., von der Grün, T. (Nov. 2003). "Body Area Network BAN--a key infrastructure element for patient-centered medical applications," *Biomed Tech (Berl)*. **47**(s1a):365-368.

Scott, K. and Bambos, N. (1996). "Routing and channel assignment for low power transmission in PCS." *International Conference on 5th IEEE Universal Personal Communications IEEE, Record, 29 Sep. - 2 Oct. 1996*. **2**:498-502.

Shen, Z.-W. Zhu, Y-H., Tian, X-Z. and Tang, Y-P. (2008). "An Ant Colony System Based Energy Prediction Routing Algorithms for Wireless Sensor Networks." *4th International Conference on Wireless Communications, Networking and Mobile Computing WiCOM '08, Oct*. 2008. 1-4.

Shih, E., Cho, S-H., Ickes, N., Min, R., Sinha, A., Wang, A. and Chandrakasan, A. (2001). "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks." *Proceedings of the 7th annual international conference on Mobile computing and networking*. Rome, Italy, ACM**.** 272-287.

Shokrzadeh, H. Haghighat, A.T., Tashtarian, F. and Nayebi, A. (2007). "Directional rumor routing in wireless sensor networks." *Internet, ICI 3rd IEEE/IFIP International Conference in Central Asia, 26-28 Sept. 2007*. 1-5.

Sinha, A. and Chandrakasan, A. (2000). "Energy Aware Software " *Proceedings of the 13th International Conference on VLSI Design, January 2000, Calcutta, India*. 50-55.

Sinha, J. D. and Barman. S. (2012). "Energy efficient routing mechanism in Wireless Sensor Network." *1st International Conference on Recent Advances in Information Technology (RAIT), 15-17 March 2012*. 300-305.

Stankovic, J. A. (2006). "*Wireless Sensor Networks*." In: Lee I., Leung J. Y-T., and Son, A. H. ed. Handbook of Real-Time and Embedded Systems. Boca Raton Florida: Chapman and Hall, 20-1 – 20-9.

Tian, H and Matsuzawa, T. (2005). "RandomWalk Routing for Wireless Sensor Networks." *International Conference on Sixth Parallel and Distributed Computing, Applications and Technologies*, 05-08 Dec. 2005. 196-200.

Touray, B and Johnson, P. (June 2012a). "Biased random algorithm for load balancing in Wireless Sensor Networks (BRALB)". *International Journal of Scientific & Engineering Research*. **3**(6).

Touray, B. and Johnson, P. (July, 2012b). "BIASED RANDOM ENERGY EFFICIENT ROUTING ALGORITHM (BREERA)." *International Journal of Engineering Science and Technology*. **4**(6):2720 - 2729.

Touray, B. Lau, J and Johnson, P. (2013). “Clustering-Biased Random Algorithm for Load Balancing (C-BRALB) in Wireless Sensor Networks”. *Int. J. Intelligent Systems Technologies and Applications*. Accepted and to be publish very soon.

Touray, B., Johnson, P. and Randels, M. (2011). "Statistical analysis of Biased Random Energy-Efficient Routing Algorithm (BREERA)." *Workshop on Next Generation Wireless Networks (WoNGeN).*

Touray, B., Shim, J. Johnson, P. (Sept.2012c). “Biased Random Algorithm for Load Balancing in Wireless Sensor Networks (BRALB)”, *15th International Power Electronics and Motion Control Conference, EPE-PEMC 2012 ECCE Europe, 4-6 Sept. 2012, Novi Sad,Serbia*. LS4e.1-1,LS4e.1-5.

Tovar, A., Friesen, T. , Ferens, K. and McLeod, B. (2010.). "A DTN wireless sensor network for wildlife habitat monitoring." 23rd Canadian *Conference on Electrical and Computer Engineering (CCECE), 2-5 May 2010*. 1-5.

Vergados, D. J., Pantazis, N. A. and Vergados, D. D. (2008). "Energy-efficient route selection strategies for wireless sensor networks." *Mob. Netw. Appl*. **13**(3-4): 285-296.

Vergados, D. J., Pantazis, N.A. and Vergados, D.D. (2008). "Energy-efficient route selection strategies for wireless sensor networks." *Mob. Netw. Appl*. **13**(3-4): 285-296.

Wang, D. and Qi, X. (2008). "An Algorithm for Dynamic Routing Based on Multi-paths in Wireless Sensor Networks." *International Conference on Computer Science and Software Engineering, 12-14 Dec. 2008* . 536-539.

Wang, X. Xing, G., Zhang, Y., Lu, C., Pless, R. and Gill, C. (2003). "Integrated coverage and connectivity configuration in wireless sensor networks." *SenSys '03 Proceedings of the 1st international conference on Embedded networked sensor systems*. New York, NY, USA: ACM. 28-39.

Wei, X. and Zhi, L. (2010). "The Multi-Objective Routing Optimization of WSNs Based on an Improved Ant Colony Algorithm." *6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), 23-25 Sept. 2010*. 1-4.

Weste, N. H. E. and Eshraghian, K. (1993). "*Principles of CMOS VLSI Design, A Systems Perspective, 2nd edition*." Reading Mass: Addison-Wesley Longman.

Wu, S.-L. and Tseng, Y.-C. (2007). "*Wireless ad hoc networking : personal-area, local-area, and the sensory-area networks*." Boca Raton, New York: Auerbach Publications.

Xu, Y., Heidemann, J. and Estrin, D. (2001). "Geography-informed energy conservation for Ad Hoc routing." *Proceedings of the 7th annual international conference on Mobile computing and networking, Rome, Italy*. New York, NY, USA: ACM**.** 70-84.

Yan, T., He, T. and Stankovic, J. A. (2003). "Differentiated Surveillance Service for Sensor Networks." *SenSys'03*. 51-62.

Yanrong, C., and Cao, J. (2007). "An Improved Directed Diffusion for Wireless Sensor Networks." *International Conference on Wireless Communications, Networking and Mobile Computing, WiCom 2007, 21-25 Sept. 2007*. 2380-2383.

Ye, F. Chen, A., Lu, S., Zhang, L. (2001). "A scalable solution to minimum cost forwarding in large sensor networks." *In Proceedings of the tenth International Conference on computer Communications and Networks (ICCN)*. 304-309.

Ye, F., Zhong, G., Lu, S., Zhang, L. (March 2005). "GRAdient Broadcast: A Robust Data Delivery Protocol for Large Scale Sensor Networks." *Wireless Networks(WINET)*. **11**(2): 285-298.

Younis, O. and Fahmy, S. (2004). "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks." *IEEE Transactions on Mobile Computing*. **3**(4): 366-379.

Zhang, H. and Hou, J. (2005). "Maintaining sensing coverage and connectivity in large sensor networks." *Ad hoc & Sensor Wireless Networks*. **1**(1-2).

ZigBee Alliance. (2003). "ieee802.15.4-2003." Available: http://www.zigbee.org. Last accessed 25th April 2013.

Zimmerman, T. G. (1996). "Personal Area Networks: Near-field intrabody communication." *IBM Systems Journal* . **35**(3.4): 609-617.

Appendices

Appendix A

List of Authors' Publications

Journal Publications

(1) Barra Touray and P. Johnson, (6^{th} June 2012). "Biased random algorithm for load balancing in Wireless Sensor Networks (BRALB)". *International Journal of Scientific & Engineering Research*. 3(6).

(2) Barra Touray and P. Johnson, (June 2012). Biased Random Energy-Efficient Routing Algorithm (BREERA*). International Journal of Engineering Science and Technology (IJEST.4*

(3) Barra Touray, Jie Lau and P. Johnson, (2013). "Clustering-Biased Random Algorithm for Load Balancing (C-BRALB) in Wireless Sensor Networks". *Int. J. Intelligent Systems Technologies and Applications.* Accepted and to be publish very soon.

International Conferences Publications

(1) B. Touray, P. Johnson, and M. Randels, (2011). "Statistical Analysis of Biased Random Energy-Efficient Routing Algorithm (BREERA)," *Workshop on Next Generation Wireless Networks (WoNGeN),2011*

(2) Barra Touray, Jinn Shim, and P. Johnson, (2012). "Biased Random Algorithm for Load Balancing in Wireless Sensor Networks (BRALB)" , 15th International Power Electronics and Motion Control Conference, EPE-PEMC 2012 ECCE Europe, Novi Sad,Serbia, 2012

Appendix B

Simulation Code

BRALB

```
globals [rcd-msg pdf hops]        ; declaration

turtles-own [energy]

breed [ones one]

ones-own [location]

breed [twos two]

twos-own [location]

breed [threes three]

threes-own [location]

breed [fours four]

fours-own [location]

breed [fives five]

fives-own [location]

breed [sixes six]

sixes-own [location]

breed [sevens seven]

sevens-own [location]

breed [eights eight]

eights-own [location]

breed [nines nine]

nines-own [location]

breed [tens ten]

tens-own [location]

breed [elevens eleven]

elevens-own [location]

breed [twelves twelve]

twelves-own [location]

breed [thirteens thirteen]

thirteens-own [location]
```

```
breed [fourteens fourteen]

fourteens-own [location]

breed [fifteens fifteen]

fifteens-own [location]

breed [sixteens sixteen]

sixteens-own [location]

breed [seventeens seventeen]

seventeens-own [location]

breed [eighteens eighteen]

eighteens-own [location]

breed [nineteens nineteen]

nineteens-own [location]

breed [twentys twenty]

twentys-own [location]

breed [twentyones twentyone]

twentyones-own [location]

breed [twentytwos twentytwo]

twentytwos-own [location]

breed [twentythrees twentythree]

twentythrees-own [location]

breed [twentyfours twentyfour]

twentyfours-own [location]

breed [twentyfives twentyfive]

twentyfives-own [location]

to setup                        ; for setting up the wireless sensor environment
  clear-all
   set-default-shape turtles "square"
    ask patches with [ abs pxcor < (grid-size / 2  ) and abs pycor < (grid-size / 2 )]
       [sprout 1
      ]
       ask turtles [
       create-links-with turtles-on neighbors4
       ask links [set color blue ]
       set energy (energy + nodes-energy)
       set label energy
```

```
    ]
    ask turtles [
   setxy (xcor * (max-pxcor - 1  ) / (grid-size / 2 - 0.5 ))
         (ycor * (max-pycor - 1 ) / (grid-size / 2 - 0.5 ))
    set color red
     ]
   reset-ticks
end
to set-source1                    ; for setting up the source node within the network
  create-ones messages [           ; for assigning the messages at source
  set color red

  set size 2

  set location turtle source1

  move-to location

  ask turtles-here [set color gray]

]

end



to set-source2                    ; for setting up the source node within the network

 create-twos messages [            ; for assigning the messages at source

  set color red

  set size 2

  set location turtle source2

  move-to location

  ask turtles-here [set color red]

]

end



to set-source3                    ; for setting up the source node within the network

 create-threes messages [          ; for assigning the messages at source

  set color red

  set size 2
```

```
  set location turtle source3
  move-to location
  ask turtles-here [set color orange]
]
end

to set-source4                 ; for setting up the source node within the network
 create-fours messages [       ; for assigning the messages at source
  set color red
  set size 2
  set location turtle source4
  move-to location
  ask turtles-here [set color brown]
]
end

to set-source5                 ; for setting up the source node within the network
 create-fives messages [       ; for assigning the messages at source
  set color red
  set size 2
  set location turtle source5
  move-to location
  ask turtles-here [set color yellow]
]
end

to set-source6                 ; for setting up the source node within the network
```

```
 create-sixes messages [              ; for assigning the messages at source
   set color red
   set size 2
   set location turtle source6
   move-to location
   ask turtles-here [set color cyan]
]
end

to set-source7                        ; for setting up the source node within the network
 create-sevens messages [             ; for assigning the messages at source
   set color red
   set size 2
   set location turtle source7
   move-to location
   ask turtles-here [set color turquoise]
]
end

to set-source8                        ; for setting up the source node within the network
 create-eights messages [              ; for assigning the messages at source
   set color red
   set size 2
   set location turtle source8
   move-to location
   ask turtles-here [set color cyan]
]
```

```
end

to set-source9                         ; for setting up the source node within the network
 create-nines messages [               ; for assigning the messages at source
   set color red
   set size 2
   set location turtle source9
   move-to location
   ask turtles-here [set color sky]
]
end


to set-source10                        ; for setting up the source node within the network
 create-tens messages [                ; for assigning the messages at source
   set color red
   set size 2
   set location turtle source10
   move-to location
   ask turtles-here [set color blue]
]
end

to set-source11                        ; for setting up the source node within the network
 create-elevens messages [             ; for assigning the messages at source
   set color red
   set size 2
   set location turtle source11
   move-to location
```

```
  ask turtles-here [set color violet]
]
end

to set-source12                    ; for setting up the source node within the network
 create-twelves messages [          ; for assigning the messages at source
  set color red
  set size 2
  set location turtle source12
  move-to location
  ask turtles-here [set color magenta]
]
end

to set-source13                    ; for setting up the source node within the network
 create-thirteens messages [        ; for assigning the messages at source
  set color red
  set size 2
  set location turtle source13
  move-to location
  ask turtles-here [set color pink]
]
end

to set-source14                    ; for setting up the source node within the network
 create-fourteens messages [        ; for assigning the messages at source
  set color red
  set size 2
```

```
  set location turtle source14
  move-to location
  ask turtles-here [set color cyan]
]
end

to set-source15                    ; for setting up the source node within the network
 create-fifteens messages [        ; for assigning the messages at source
  set color red
  set size 2
  set location turtle source15
  move-to location
  ask turtles-here [set color pink]
]
end

to set-source16                    ; for setting up the source node within the network
 create-sixteens messages [        ; for assigning the messages at source
  set color red
  set size 2
  set location turtle source16
  move-to location
  ask turtles-here [set color magenta]
]
end


to set-source17                    ; for setting up the source node within the network
```

```
 create-seventeens messages [             ; for assigning the messages at source
   set color red
   set size 2
   set location turtle source17
   move-to location
   ask turtles-here [set color red]
]
end

to set-source18                        ; for setting up the source node within the network
 create-eighteens messages [             ; for assigning the messages at source
   set color red
   set size 2
   set location turtle source18
   move-to location
   ask turtles-here [set color yellow]
]
end

to set-source19                        ; for setting up the source node within the network
 create-nineteens messages [             ; for assigning the messages at source
   set color red
   set size 2
   set location turtle source19
   move-to location
   ask turtles-here [set color blue]
]
end
```

```
to set-source20                 ; for setting up the source node within the network
 create-twentys messages [          ; for assigning the messages at source
   set color red
   set size 2
   set location turtle source20
   move-to location
   ask turtles-here [set color violet]
]
end

to set-source21                 ; for setting up the source node within the network
 create-twentyones messages [          ; for assigning the messages at source
   set color red
   set size 2
   set location turtle source21
   move-to location
   ask turtles-here [set color blue]
]
end

to set-source22                 ; for setting up the source node within the network
 create-twentytwos messages [          ; for assigning the messages at source
   set color red
   set size 2
   set location turtle source22
   move-to location
```

```
  ask turtles-here [set color orange]
]
end

to set-source23                    ; for setting up the source node within the network
 create-twentythrees messages [            ; for assigning the messages at source
  set color red
  set size 2
  set location turtle source23
  move-to location
  ask turtles-here [set color cyan]
]
end

to set-source24                    ; for setting up the source node within the network
 create-twentyfours messages [            ; for assigning the messages at source
  set color red
  set size 2
  set location turtle source24
  move-to location
  ask turtles-here [set color brown]
]
end

to set-source25                    ; for setting up the source node within the network
 create-twentyfives messages [            ; for assigning the messages at source
  set color red
  set size 2
```

```
   set location turtle source25
   move-to location
   ask turtles-here [set color gray]
]
end

to set-destination                        ; for setting up the destination node within the network
  ask turtle destination [
    set color green
    set size 2
  ]
end

to go                                     ; for executing the algorithm
  ask ones [
    let new-location max-one-of [link-neighbors] of location [energy]
    ask [link-with new-location] of location [ set thickness 0.5 ]
    ask turtles-here [ set energy energy - 1  ; making the nodes energy to decrease by one
unit for every message
    set label energy]
    move-to new-location
    set location new-location
    set hops (hops + 1)
     if new-location = turtle destination [
    set rcd-msg (rcd-msg + 1)
    die ]
     if ticks >= HopsToLive [die]
```

```
]

ask twos [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]
]

ask threes [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
```

```
  die ]
    if ticks >= HopsToLive [die]

]

ask fours [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]

]

ask fives [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
```

```
  if new-location = turtle destination [
 set rcd-msg (rcd-msg + 1)
 die ]
   if ticks >= HopsToLive [die]

]

ask sixes [
 let new-location max-one-of [link-neighbors] of location [energy]
 ask [link-with new-location] of location [ set thickness 0.5 ]
 ask turtles-here [ set energy energy - 1
 set label energy]
 move-to new-location
 set location new-location
 set hops (hops + 1)
  if new-location = turtle destination [
 set rcd-msg (rcd-msg + 1)
 die ]
   if ticks >= HopsToLive [die]

]


ask sevens [
 let new-location max-one-of [link-neighbors] of location [energy]
 ask [link-with new-location] of location [ set thickness 0.5 ]
 ask turtles-here [ set energy energy - 1
 set label energy]
 move-to new-location
```

```
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]

]

ask eights [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]

]

ask nines [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
```

```
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]

]

ask tens [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]

]

ask elevens [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
```

```
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]

]

ask twelves [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]

]

ask thirteens [
```

```
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]

]

ask fourteens [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]

]
```

```
ask fifteens [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]

]

ask sixteens [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]
```

```
]

ask seventeens [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]
]

ask eighteens [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
```

```
  die ]
    if ticks >= HopsToLive [die]

]

ask nineteens [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]

]

ask twentys [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
```

```
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]

]

ask twentyones [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]

]


ask twentytwos [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
```

```
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]

]

ask twentythrees [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]

]


ask twentyfours [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
```

```
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]

]


ask twentyfives [
  let new-location max-one-of [link-neighbors] of location [energy]
  ask [link-with new-location] of location [ set thickness 0.5 ]
  ask turtles-here [ set energy energy - 1
  set label energy]
  move-to new-location
  set location new-location
  set hops (hops + 1)
   if new-location = turtle destination [
  set rcd-msg (rcd-msg + 1)
  die ]
    if ticks >= HopsToLive [die]
]

tick
  end
```

C-BRALB

```
reed [messages message]
Globals
  [
  mgf
  rich-nbr
  chead2
  link-ch
  generated
  delivered
  failed
  en
  te
  te2
  ]
turtles-own [ energy snk clustered ch cluster nn m reported]
links-own [visited]
messages-own [location]
;;;;;;;;;;;;;;;;;;
;;;;;;SETUP;;;;;;
;;;;;;;;;;;;;;;;;;


to setup
    ;; (for this model to work with NetLogo's new plotting features,
  ;; __clear-all-and-reset-ticks should be replaced with clear-all at
  ;; the beginning of your setup procedure and reset-ticks at the end
  ;; of the procedure.)
  __clear-all-and-reset-ticks ;clear all
    set delivered 0
    set generated 0
    set failed 0
    make-nodes ;refer to make turtle
    set te 0

end


to makesink
      ask turtle sink
     [
      set color orange
      set size 2
      set snk 1
      set energy 0
    ]
end


to make-nodes
    set-default-shape turtles "circle"
```

```
  crt num-nodes + 1
  ;set-default-shape turtles "circle"
  ask turtles
  [
    set m 0
    set energy 1000
    setxy (random-xcor * 0.95) (random-ycor * 0.95) ; this will help to keep nodes
away from edges
    set color white
    layout-spring turtles links 0.1 (world-width / (sqrt num-nodes)) 1 ;keep nodes away
from each others
    ;set label energy
  ]

end



;;;;;;;;;;;;;;;;;
;;;Generate Msg;;;
;;;;;;;;;;;;;;;;;

to gm

ask turtle Message-Generate-From
[
 set color red
 set ch 1
 set energy energy - 1
 show energy
]
 ; setup-cluster1
 ;find-neighborhood



end

;;;;;;;;;;;;;;;;;;;
;;;;Make Cluster;;;;
;;;;;;;;;;;;;;;;;;;

to clustering
 let c one-of turtles with [color = red and ch = 1]

 ifelse c != nobody
 [
  ask c
  [

   let nbors turtles in-radius Range-of-Nodes
   let members nbors with [m = 0]
   let sinknode one-of members with [ snk = 1 ]
   if sinknode != nobody
```

```
    [
      set ch 2
      set color blue
      set m 1
      create-link-with sinknode
      set energy energy - 1
      show energy
      ;ask sinknode [set color sky]
      show "Message Recieved To Sink Please Regenerate Message Again Or Press
Clustering Button to Auto Generate Message"
      stop
    ]
    create-links-with other members with [ m = 0 ]
    ;ask my-links [ set visited true]
    ask link-neighbors
    [
      if m = 0
      [
      set color green
      set m 1
      set energy energy - 1
      show energy
      ]
    ]

   ; Properties of cluster Head are
    set m 1
    set ch 2
    set color blue
    set energy energy - 1
    show energy
    ;selecting new clusterhead
    let d members with [ ch = 0 ]
    let dd max-one-of d [distance myself]
    ifelse dd != nobody
    [
      ask dd
      [

        set color red
        set ch 1
      ]
    ]
    [
      show "All Nodes are Clustered here Please Regenerate Message"
      stop
    ]
  ]
 ]

 ;else

 [
```

```
    let new one-of turtles with [color = white]
    ifelse new != nobody
    [
      ask new
      [
        set color red
        set ch 1
        set energy energy - 1
        show energy
      ]
    ]
    [
      show " All nodes are clustered or no node have any information"
      beep
      stop
    ]
  ]
end

;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;

to clusterhead-link
let chead one-of turtles with [ch = 2]
let range2 range-of-nodes * 2
ifelse chead != nobody
[
  ask chead
  [
    set ch 3
    let nbr turtles in-radius range2 ;nbr=neighbors
    let nchead nbr with [ch = 2] ;nchead =neighbor cluster-heads
    let snbr one-of nbr with [snk = 1] ; snbr=sink in neighborhood
    set energy energy - 1
    show energy
    ifelse snbr != nobody
    [
      create-link-with snbr
    ]
    [
      create-links-with nchead
      ask nchead
      [
        set energy energy - 1
        show energy
      ]

    ]
  ]
]
[
  show " All possible cluster-heads connected with eachother"
  stop
```

```
]
end

;;;;;;;;;;;;;;;;;;;;;;;;;;
;;Generate New Message;;
;;;;;;;;;;;;;;;;;;;;;;;;;;
to gnm
  ask turtle Message-Generate-From
  [
  Set size 3
  set shape "default"
  set energy energy - 1
  show energy
  set mgf turtle message-generate-from
  set generated generated + 1
  ]
  reset-ticks
end

;;;;;;;;;;;;;;;;;;
;;Move Message;;;
;;;;;;;;;;;;;;;;;;
to go
  if ticks >= Threshold
  [
    ask turtles with [ snk = 0 ]
    [
      set size 1
      set shape "circle"
    ]
    show "Message Sending Failed"
    set failed failed + 1
    repeat 2 [ beep wait 0.6 ]
    stop
  ]
  if mgf != nobody ;if there is someone
  [
    ask mgf
    [
      if color = blue
      [
        ch-got-msg
        set mgf rich-nbr
        ;show mgf
      ]
      if color = green
      [
        mmbr-got-msg
        set mgf link-ch
      ]
    ]
  ]
  tick
```

```
end

to ch-got-msg
  ask mgf
  [
    let link-nbrs link-neighbors
    let sink-nbr one-of link-nbrs with [color = orange]
    ;show sink-nbr
    ifelse sink-nbr != nobody
    [
      show "Message Delivered to Sink Node"
      show "Please Generate New Message"
      set delivered delivered + 1
      set size 1
      set shape "circle"
      set energy energy - 1
      show energy
      repeat 3 [ beep wait 0.5 ]
    ]
    [

      let ch-nbrs link-nbrs with [color = blue]
      set rich-nbr max-one-of ch-nbrs [ energy ]

      ask rich-nbr
      [
        set size 3
        set shape "default"
        set energy energy - 1
        show energy
      ]
      set size 1
      set shape "circle"
      set energy energy - 1
      show energy
    ]

  ]
end

to mmbr-got-msg
  ask mgf
  [
    set link-ch link-neighbors
    ask link-neighbors
    [
      set size 3
      set shape "default"
      set energy energy - 1
      show energy
    ]
    set energy energy - 1
    show energy
```

```
    set size 1
    set shape "circle"
  ]
end

to test
    let unreported one-of turtles with [reported = 0]
    ifelse unreported != nobody
    [
      ask unreported
      [
        set en energy
        set reported 1

      ]
      set te te + en
      set te2 te
    ]
    [
      set te 0
      ask turtles[set reported 0]
      stop
    ]

end
```